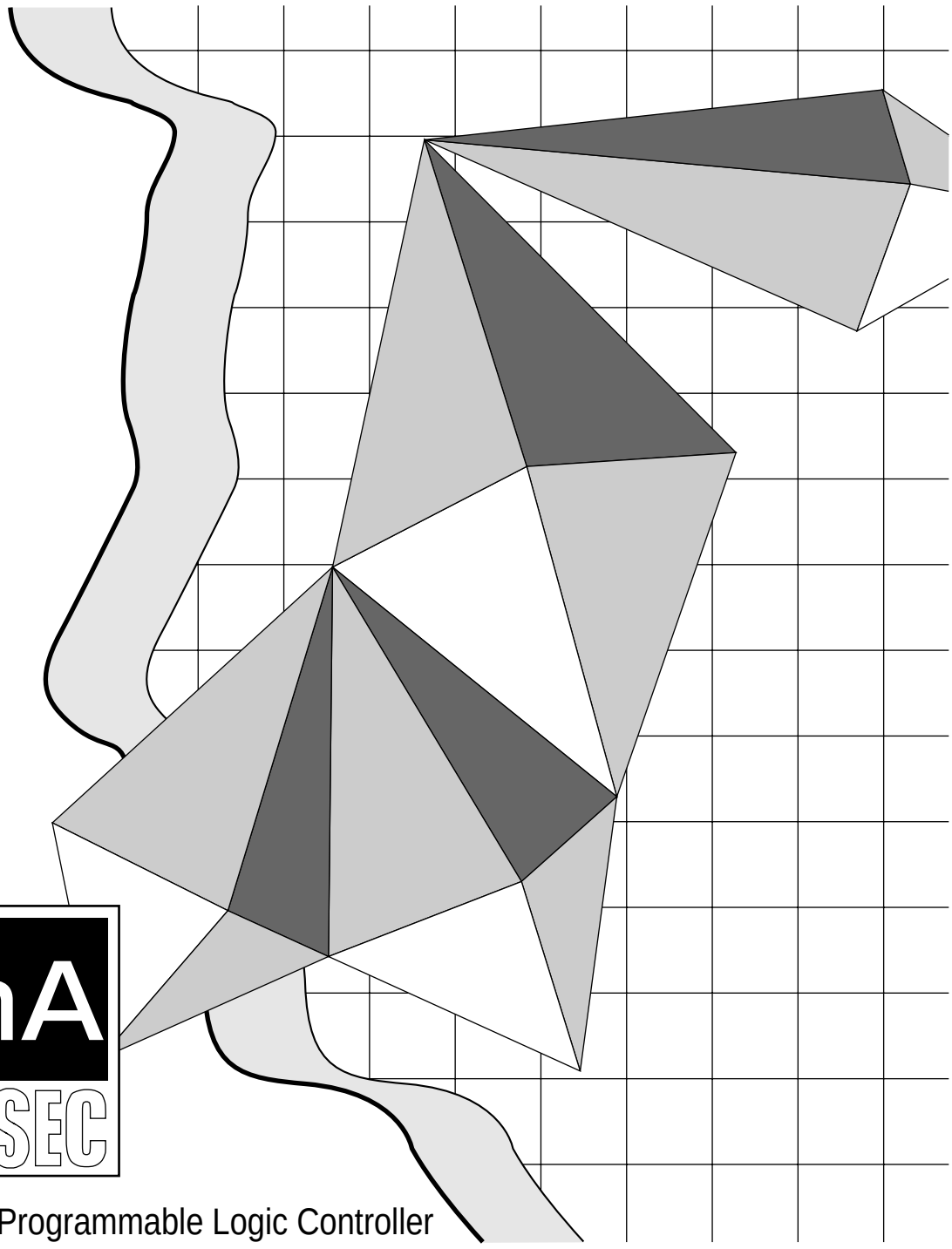
MITSUBISHI

QnA SERIES

Model Q2AS (H) CPU (S1)

User's Manual



QnA

MELSEC

Mitsubishi Programmable Logic Controller

● SAFETY PRECAUTIONS ●

(Read these precautions before using.)

When using Mitsubishi equipment, thoroughly read this manual and the associated manuals introduced in this manual.

These ● SAFETY PRECAUTIONS ● classify the safety precautions into two categories: "DANGER" and "CAUTION".

⚠ DANGER	**Procedures which may lead to a dangerous condition and cause death or serious injury if not carried out properly.**
⚠ CAUTION	**Procedures which may lead to a dangerous condition and cause superficial to medium injury, or physical damage only, if not carried out properly.**

Depending on circumstances, procedures indicated by ⚠ CAUTION may also be linked to serious results.

In any case, it is important to follow the directions for usage.

Store this manual in a safe place so that you can take it out and read it whenever necessary. Always forward it to the end user.

[DESIGN PRECAUTIONS]

⚠ DANGER

- Install a safety circuit external to the PC that keeps the entire system safe even when there are problems with the external power supply or the PC module. Otherwise, trouble could result from erroneous output or erroneous operation.

  (1) Outside the PC, construct mechanical damage preventing interlock circuits such as emergency stop, protective circuits, positioning upper and lower limits switches and interlocking forward /reverse operations.

  (2) When the PC detects the following problems, it will stop calculation and turn off all output.
  - The power supply module has over current protection equipment and over voltage protection equipment.
  - The PC CPUs self-diagnostic functions, such as the watchdog timer error, detect problems. In addition, all output will be turned on when there are problems that the PC CPU cannot detect, such as in the I/O controller. Build a fail safe circuit exterior to the PC that will make sure the equipment operates safely at such times. See Section 19.1 of this user's manual for example fail safe circuits.

  See this user's manual for example fail safe circuits.

  (3) Output could be left on or off when there is trouble in the outputs module relay or transistor. So build an external monitoring circuit that will monitor any single outputs that could cause serious trouble.

[DESIGN PRECAUTIONS]

⚠ DANGER

- When overcurrent which exceeds the rating or caused by short-circuited load flows in the output module for a long time, it may cause smoke or fire. To prevent this, configure an external safety circuit, such as fuse.
- Build a circuit that turns on the external power supply when the PC main module power is turned on. If the external power supply is turned on first, it could result in erroneous output or erroneous operation.
- When there are communication problems with the data link, the communication problem station will enter the following condition.
  Build an interlock circuit into the PC program that will make sure the system operates safely by using the communication state information. Not doing so could result in erroneous output or erroneous operation.
  (1) For the data link data, the data prior to the communication error will be held.
  (2) The MELSECNET (II,/B,/10) remote I/O station will turn all output off.
  (3) The MELSECNET/MINI-S3 remote I/O station will hold the output or turn all output off depending on the E.C. remote setting.
  Refer to the data link manuals regarding the method for setting the communication problem station and the operation status when there are communication problem.
- When performing control (data change) for the operating PLC where a peripheral device is connected to the CPU module or a personal computer, etc. is connected to the special function module, configure up an interlock circuit in the sequence program so that the whole system will always operate safely.
  When performing the other control (program change, operating status change (status control)) for the operating PLC, read the manual carefully and fully ensure safety before starting that control.
  Especially in the above control performed for the remote PLC from an external device, immediate troubleshooting of PLC side problems may not be done due to data communication error.
  Configure up an interlock circuit in the sequence program, and also determine between the external device and PLC CPU the system remedies, etc. to be taken at occurrence of data communication error.
- When configuring a system, do not leave any slots vacant on the base. Should there be any vacant slots, always use a blank cover (A1SG60) or dummy module (A1SG62).When the extension base A1S52B, A1S55B or A1S58B is used, attach the dustproof cover supplied with the product to the module installed in slot 0.If the cover is not attached, the module's internal parts may be dispersed when a short-circuit test is performed or overcurrent/overvoltage is accidentally applied to the external I/O area.

⚠ CAUTION

- Do not bunch the control wires or communication cables with the main circuit or power wires, or install them close to each other. They should be installed 100 mm (3.94 inch) or more from each other. Not doing so could result in noise that would cause erroneous operation.
- When controlling items like lamp load, heater or solenoid valve using an output module, large current (approximately ten times greater than that present in normal circumstances) may flow when the output is turned OFF → ON. Take measures such as replacing the module with one having sufficient rated current.

[INSTALLATION PRECAUTIONS]

⚠ CAUTION

- Use the PLC in an environment that meets the general specifications contained in this manual. Using this PLC in an environment outside the range of the general specifications could result in electric shock, fire, erroneous operation, and damage to or deterioration of the product.
- Install so that the pegs on the bottom of the module fit securely into the base unit peg holes, and use the specified torque to tighten the module's fixing screws. Not installing the module correctly could result in erroneous operation, damage, or pieces of the product falling.
- When installing more cables, be sure that the base unit and the module connectors are installed correctly. After installation, check them for looseness. Poor connections could cause an input or output failure.
- Correctly connect the memory card installation connector to the memory card. After installation, be sure that the connection is not loose. A poor connection could cause an operation failure.
- Always switch off all phases of the external power supply before mounting or dismounting the module. Not doing so could cause damage to the product.
- Do not directly touch the module's conductive parts or electronic components. Touching the conductive parts could cause an operation failure or give damage to the module.

[WIRING PRECAUTIONS]

⚠ DANGER

- Completely turn off the external power supply when installing or placing wiring. Not completely turning off all power could result in electric shock or damage to the product.
- When turning on the power supply or operating the module after installation or wiring work, be sure that the module's terminal covers are correctly attached. Not attaching the terminal cover could result in electric shock.

CAUTION

- Be sure to ground the FG terminals and LG terminals to the protective ground conductor. Not doing so could result in electric shock or erroneous operation.
- When wiring in the PLC, be sure that it is done correctly by checking the product's rated voltage and the terminal layout. Connecting a power supply that is different from the rating or incorrectly wiring the product could result in fire or damage.
- Do not connect multiple power supply modules in parallel. Doing so could cause overheating, fire or damage to the power supply module. If the terminal screws are too tight, it may cause falling, short circuit or erroneous operation due to damage of the screws or module.

[WIRING PRECAUTIONS]

CAUTION

- Tighten the terminal screws with the specified torque. If the terminal screws are loose, it could result in short circuits, fire, or erroneous operation.
  Tightening the terminal screws too far may cause damages to the screws and/or the module, resulting in fallout, short circuits, or malfunction.
- Be sure there are no foreign substances such as sawdust or wiring debris inside the module. Such debris could cause fires, damage, or erroneous operation.
- External connections shall be crimped or pressure welded with the specified tools, or correctly soldered. For information regarding the crimping and pressure welding tools, see the I/O module's user's manual. Imperfect connections could result in short circuit, fires, or erroneous operation.

[STARTUP AND MAINTENANCE PRECAUTIONS]

DANGER

- Do not touch the terminals while power is on. Doing so could cause shock or erroneous operation.
- Correctly connect the battery. Also, do not charge, disassemble, heat, place in fire, short circuit, or solder the battery. Mishandling of battery can cause overheating or cracks which could result in injury and fires.
- Switch all phases of the external power supply off when cleaning the module or tightening the terminal screws and module mounting screws. Not doing so could result in electric shock. If the screws are too tight, it may cause falling, short circuit or erroneous operation due to damage of the screws or modules.

CAUTION

- The online operations conducted for the CPU module being operated, connecting the peripheral device (especially, when changing data or operation status), shall be conducted after the manual has been carefully read and a sufficient check of safety has been conducted. Operation mistakes could cause damage or problems with of the module.
- Do not disassemble or modify the modules. Doing so could cause trouble, erroneous operation, injury, or fire.
- When using a wireless communication device, such as a cellular phone or PHS (Personal Handy Phone System), separate it at least 25cm away from the PLC.
- Switch all phases of the external power supply off before mounting or removing the module. If you do not switch off the external power supply, it will cause failure or malfunction of the module.
- Do not drop or give impact to the battery fitted to the module. Doing so may damage the battery, causing battery liquid leakage in the battery.
  Dispose of the battery that has been dropped or given impact, without using it.

[DISPOSAL PRECAUTIONS]

CAUTION

- When disposing of this product, treat it as industrial waste.

Revisions

* The manual number is noted at the lower left of the back cover.

Print Date	*Manual Number	Revision
Sep. 1996	SH (NA)-3599-A	First printing
Feb. 1999	SH(NA)-3599-B	Additional model SW□5□-GPPW GPP Function Software Package for Windows Addition The contents of the function version B has been added. Section 2.2, Section 7.2, Sections 8.2.1, 8.2.2, Section 19.8, Appendix 7, 8 Partial correction Safety Precautions, Contents, Section 1.2, Section 3.3.1, Chapter 4, Section 5.3, Section 6.1, Section 8.10.1, Section 15.2, Section 16.1, Section 19.7.1, Section 21.3.1, Appendix 1.6, Appendix 2
Dec., 2002	SH(NA)-3599-C	Equivalent to Japanese version D Partial correction SAFETY PRECAUTIONS, Section 3.1.2, Section 3.3.1, Section 3.3.2, Section 3.3.3, Section 8.2.1, Section 8.5, Section 12.1.5, Section 12.1.6, Chapter 13, Section 14.3, Section 15.1, Section 15.3, Section 16.1, Section 16.2, Section 17.2, Section 17.3, Section 19.1, Section 22.3.2, Appendix 3

This manual does not imply guarantee or implementation right for industrial ownership or implementation of other rights. Mitsubishi Electric Corporation is not responsible for industrial ownership problems caused by use of the contents of this manual.

© 1996 Mitsubishi Electric Corporation

INTRODUCTION

Thank you for choosing a Mitsubishi MELSEC-QnA Series General Purpose Programmable Controller.
Before using your new PC, please read this manual thoroughly to gain an understanding of its functions so you can use it properly.
Please forward a copy of this manual to the end user.

Table of Contents

1 Using This Manual ..... 1-1 to 1-2

1.1 Using This Manual 1-1
1.2 Abbreviations and General Names Used in This Manual 1-2

2 Overview ..... 2-1 to 2-7

2.1 Features 2-1
2.2 Q2ASCPU Additional Functions 2-4
2.2.1 Summary of Additional Functions 2-6

3 System Configuration ..... 3-1 to 3-16

3.1 System Configuration 3-1
3.1.1 Independent System Machine Configuration 3-1
3.1.2 Q2ASCPU Preipheral Device Configurations 3-2
3.2 System Configuration Overview 3-3
3.3 List of System Equipment 3-5
3.3.1 List of System Equipment 3-5
3.3.2 Precaution on System Configuration 3-13
3.3.3 Q2ASCPU Memory Block Diagram 3-16

4 Performance Specification ..... 4-1 to 4-2

5 Allocation of I/O Numbers ..... 5-1 to 5-11

5.1 What are I/O Numbers? 5-1
5.2 I/O Number Allocation Scheme 5-2
5.3 I/O Allocation by GPP Function 5-4
5.4 I/O Number Allocation Examples 5-8

6 Data Exchange with Special Function Modules ..... 6-1 to 6-3

6.1 Reading and Writing Data from Q2ASCPU by FROM/TO Instruction 6-1
6.2 Reading and Writing Data from Q2ASCPU by Special Direct Device 6-2
6.3 Processing a Data-Communication Request from the Special Function Module 6-3

7 Setting Automatic Refresh ..... 7-1 to 7-10

7.1 For MELSECNET/MINI-S3 ..... 7.1
7.2 CC-Link Automatic Refresh Setting 7.5

8 Debug Function ..... 8-1 to 8-57

8.1 Function List 8-1
8.2 Monitoring Function 8-2
8.2.1 Monitor Condition Setting 8-2
8.2.2 Monitoring Test for Local Device 8-10
8.3 Writing During RUN 8-13
8.4 Execution Time Measurement 8-15
8.4.1 Program List Monitor 8-15

8.4.2 Interrupt Program List Monitor 8-18
8.4.3 Scan Time Measurement 8-19
8.5 Sampling Trace Function 8-21
8.6 Status Latch Function 8-31
8.7 Step Drive 8-37
8.7.1 Step Execution 8-38
8.7.2 Partial Execution 8-40
8.7.3 Step-By-Step Execution 8-43
8.8 Program Trace Function 8-44
8.9 Simulation Function 8-52
8.10 Debug Function with Multiple Users 8-55
8.10.1 Multiple-User Monitoring Function 8-56
8.10.2 Multiple-User RUN Write Function 8-57

9 Maintenance Functions ..... 9-1 to 9-16

9.1 Function List 9-1
9.2 Watchdog Timer 9-2
9.3 Self-Diagnosis Function 9-4
9.3.1 Interrupt due to Error Occurrence 9-7
9.3.2 LED Display when Error Occurs 9-7
9.3.3 Cancel Error 9-8
9.4 Failure History 9-9
9.5 System Protect 9-10
9.6 Password Registration 9-10
9.7 System Display 9-12
9.8 LED Display 9-13
9.8.1 LED Display 9-13
9.8.2 Priority Setting 9-15

10 Other Functions ..... 10-1 to 10-20

10.1 Function List 10-1
10.2 Constant Scan 10-2
10.3 Latch Functions 10-5
10.4 Setting the Output (Y) Status when Changing from STOP Status to RUN Status 10-6
10.5 Clock Function 10-7
10.6 Remote Operation 10-11
10.6.1 Remote RUN/STOP 10-11
10.6.2 Remote STEP-RUN 10-13
10.6.3 Remote PAUSE 10-14
10.6.4 Remote RESET 10-16
10.6.5 Remote Latch Clear 10-17
10.6.6 Relationship of the Remote Operation and RUN/STOP Key Switch of CPU module 10-18
10.7 Terminal Operation 10-19
10.7.1 Message Display Operation 10-19
10.7.2 Key Entry Operation 10-19
10.8 Reading the Module Access Time Interval 10-20

11 Comments That Can Be Stored in Q2ASCPU ..... 11-1 to 11-5

11.1 Function List 11-1
11.2 Name 11-1
11.3 Driver Title 11-2

11.4 File Title 11-3
11.5 Device Comments 11-3
11.6 Statement/Documentation 11-5
11.7 Device Initialization Comment 11-5

12 Overview of the Q2ASCPU Calculation Processing ..... 12-1 to 12-23

12.1 Program Execution Type Overview 12-1
12.1.1 Initial Execution Type Program 12-3
12.1.2 Scan Execution Type Program 12-5
12.1.3 Low-Speed Execution Type Program 12-7
12.1.4 Stand-by Type Program 12-12
12.1.5 Initial Processing 12-18
12.1.6 I/O Module Refresh Processing 12-18
12.1.7 END Processing 12-19
12.2 RUN, STOP, PAUSE, STEP-RUN Operation Processing 12-20
12.3 Operation Processing during Momentary Power Failure 12-22
12.4 Data Clear Processing 12-23

13 Parameter List ..... 13-1 to 13-6

14 Setting Memory Card Type ..... 14-1 to 14-2

14.1 Memory Card Usage 14-1
14.2 Difference between E^2PROM and Flash Memory 14-1
14.3 Setting Memory Card Capacity 14-2

15 CPU Module Hardware Specifications ..... 15-1 to 15-5

15.1 General Specifications 15-1
15.2 Part Name and Specifications 15-2
15.3 Switch Operation and LED Display 15-4

16 Power Supply Module ..... 16-1 to 16-6

16.1 Specifications 16-1
16.1.1 List of power supply module specifications 16-1
16.1.2 Selecting the Power Supply Module 16-3
16.2 Handling Precautions 16-4
16.3 Names of Parts and Settings 16-5

17 Base Module/Extension Cable ..... 17-1 to 17-9

17.1 Base Module Specification Table 17-1
17.1.1 Main Base Module for High-Speed Access (A1S38HB) 17-2
17.2 Extension Cable Specification 17-2
17.3 Application Standards of Extension Base Unit (A1S52B (S1), A1S55B (S1), A1S58B (S2), A55B, A58B) 17-3
17.4 Handling Precautions 17-7
17.5 Part Name 17-8

18 Memory Card, Battery ..... 18-1 to 18-6

18.1 Memory Card Specifications 18-1
18.2 Handling the Memory Card 18-2
18.3 Battery Specification (for CPU module and Memory card) 18-3
18.4 Handling Precautions 18-3
18.5 Part Name 18-4
18.6 Battery Loading (for CPU module and memory card) 18-5

18.7 Memory Card Loading/Unloading Procedures 18-6

19 Loading and Installation ..... 19-1 to 19-14

19.1 On Fail-Safe Circuit 19-1
19.2 Installation Environment 19-4
19.3 Calculating Heat Generation by PC 19-4
19.4 Base-Unit Installation 19-6
19.4.1 Precaution on Installation 19-6
19.4.2 Installation 19-7
19.5 Installation and Removal of Module 19-8
19.6 Installing and Removing the Dustproof Cover 19-10
19.7 Wiring 19-11
19.7.1 Wiring Instructions 19-11
19.7.2 Wiring to Module Terminals 19-14
19.8 Precaution when Connecting the Uninterruptive Power Supply (UPS) 19-14

20 EMC and Low-Voltage Directives ..... 20-1 to 20-10

20.1 Requirements for conformance to the EMC Directive 20-1
20.1.1 Standard applicable to the EMC Directive 20-1
20.1.2 Installation Instructions for the EMC Directive 20-2
20.1.3 Cables 20-3
20.1.4 Power supply module 20-6
20.1.5 Ferrite core 20-7
20.1.6 Noise filter (power supply line filter) 20-7
20.2 Requirement to Conform to the Low-Voltage Directive 20-8
20.2.1 Standard applied for MELSEC-QnA series PLC 20 8
20.2.2 Precautions when using the MELSEC-QnA series PLC 20-8
20.2.3 Power Supply 20-9
20.2.4 Control Box 20-9
20.2.5 Module Installation 20-10
20.2.6 Grounding 20-10
20.2.7 External Wiring 20-10

21 Maintenance and Inspection ..... 21-1 to 21-6

21.1 Daily Inspection 21-1
21.2 Periodic Inspection 21-2
21.3 Battery Replacement 21-3
21.3.1 Battery Life 21-4
21.3.2 Battery Replacement Procedure 21-5

22 Troubleshooting ..... 22-1 to 22-38

22.1 Troubleshooting Basics 22-1
22.2 Troubleshooting 22-2
22.2.1 Troubleshooting Flowchart 22-2
22.2.2 Flowchart for when "POWER" LED is Turned Off 22-3
22.2.3 Flowchart for when the "RUN" LED is Turned Off 22-4
22.2.4 When "RUN" LED is Flashing 22-4
22.2.5 Flowchart for when "ERROR" LED is On/Flashing 22-5
22.2.6 When "USER" LED is Turned On 22-6
22.2.7 Flowchart for when "BAT.ARM" LED is Turned On 22-6
22.2.8 Flowchart for when Output Load of Output Module does not Turn On 22-7
22.2.9 Flowchart for when Unable to Write a Program 22-8
22.2.10 Flowchart for when it is Unable to Perform Boot Operation from Memory Card ... 22-10
22.3 Error Code List 22-12
22.3.1 Error-Code Reading Method 22-12
22.3.2 Error Code List 22-13
22.4 Error Reset 22-33
22.5 Fault Example of I/O Module 22-34
22.5.1 Input-Circuit Fault and Corrective Action 22-34
22.5.2 Output-Circuit Fault and Corrective Action 22-36

APPENDICES ..... Appendix-1 to Appendix-84

APPENDIX 1 Instruction List Appendix-1
APPENDIX 1.1 Sequence Instructions Appendix-1
APPENDIX 1.2 Basic Instructions Appendix-3
APPENDIX 1.3 Response Instructions Appendix-9
APPENDIX 1.4 Data Link Instructions Appendix-20
APPENDIX 1.5 PID Control Instructions Appendix-21
APPENDIX 1.6 Special Function Module Instructions Appendix-22
APPENDIX 2 Special Relay List Appendix-25
APPENDIX 3 Special Register List Appendix-36
APPENDIX 4 Precaution Items for Utilizing Existing MELSEC-A Series Programs in the Q2ASCPU Appendix-52
APPENDIX 4.1 Instructions Appendix-52
APPENDIX 4.2 Devices Appendix-58
APPENDIX 4.3 Parameters Appendix-59
APPENDIX 4.4 Timer and Interrupt Counter Operation Appendix-60
APPENDIX 4.5 Sequence Program, Statement and Note Appendix-61
APPENDIX 4.6 Microcomputer Program Appendix-62
APPENDIX 4.7 Comments Appendix-63
APPENDIX 4.8 Constant Scan Function and Error Check Function Appendix-63
APPENDIX 4.9 I/O Control Method Appendix-63
APPENDIX 4.10 Data Link System Appendix-63
APPENDIX 4.11 Index Register Processing Appendix-64
APPENDIX 4.12 CHK Instruction and IX Instruction Appendix-65
APPENDIX 4.13 Accessing File Register R with an Instruction Appendix-65
APPENDIX 5 Error Code Return to Request Origin during General Data Processing Appendix-66
APPENDIX 5.1 Error Code Overall Explanation Appendix-66
APPENDIX 5.2 Description of the Errors of the Error Codes (4000H to 4FFFH) Detected by the CPU module Appendix-66
APPENDIX 6 External Dimensions Diagram Appendix-71

APPENDIX 6.1 Q2AS(H)CPU(S1) Module Appendix-71
APPENDIX 6.2 Power Supply Module (A1S61PN, A1S62PN, A1S63P) Appendix-71
APPENDIX 6.3 Main Base Module Appendix-72
APPENDIX 6.4 Extension Base Module Appendix-74
APPENDIX 7 Using Local Device Stored in the Subroutine/Interrupt Program File (function version B or later) Appendix-80
APPENDIX 8 Network Relay from Ethernet Module(function version B or later) Appendix-83

About this manual

The following are manuals related to this product.
Request for the manuals as needed according to the chart below.

Related Manuals

Manual Name	Manual No. (Type code)
QnACPU GUIDEBOOK For the first-time user of QnACPU, describes steps on creating a program, writing the program in the CPU module, and debugging. Describes usage of QnACPU features. (Sold separately)	IB-66606 (13JF10)
QnACPU PROGRAMMING MANUAL (Fundamentals) Describes programming methods, device names and program types that are necessary in program creation. (Sold separately)	IB-66614 (13JF46)
QCPU(Q mode)/QnACPU PROGRAMMING MANUAL (Common Instructions) Describes how to use sequence instructions, basic instructions and application instructions. (Sold separately)	SH-080039 (13JF58)
QnACPU PROGRAMMING MANUAL (Special Functions) Describes dedicated instructions used in special-function modules. (Sold separately)	IB-66616 (13JF48)
QnACPU PROGRAMMING MANUAL (AD57 Instructions) Describes dedicated instructions used to control the AD57(S1)-type CRT controller module. (Sold separately)	IB-66617 (13JF49)
QCPU(Q mode)/QnACPU PROGRAMMING MANUAL (PID Control Instructions) Describes dedicated instructions used for PID control in Q2ACPU(S1), Q3ACPU and Q4ACPU. (Sold separately)	SH-080040 (13JF59)
QCPU(Q mode)/QnACPU PROGRAMMING MANUAL (SFC) Describes system components, performance specifications, functions, programming, debug going and error codes of MELSAP. (Sold separately)	SH-080041 (13JF60)
AnS Module Type I/O User's Manual Describes specification of AnS module as I/O module. (Sold separately)	IB-66541 (13JE81)
Type QnA/Q4AR MELSECNET/10 Network System Reference Manual Describes MELSECNET/10 overview, specifications, part names and settings. (Sold separately)	IB-66620 (13JF77)
Type MELSECNET, MELSECNET/B Data Link System Reference Manual Describes MELSECNET(II) and MELSECNET/B overview, specifications, part names and settings. (Sold separately)	IB-66350 (13JF70)
GX Developer Version7 Operating Manual Describes the online functions of GX Developer including the programming procedure, printing out procedure, and debugging procedure. (Included with product)	SH-080166 (13JU14)
Type SW2IV-GPPQ GPP Software package OPERATING MANUAL (Offline) Describes SW2IVD-GPPQ's offline functions such as program creation, printout method and file maintenance. (Included with product)	IB-66774 (13J921)
Type SW2IVD-GPPQ GPP Software package OPERATING MANUAL (Online) Describes SW2IVD-GPPQ's online functions such as monitoring and debugging methods. (Included with product)	IB-66775 (13J922)

Type SW2IVD-GPPQ GPP Software package OPERATING MANUAL (SFC) Describes MELSAP-3 system components, performance specifications, functions, system start-up procedure, SFC program editing method, monitoring method, printout method and error messages. (Included with product)	IB-66776 (13J923)
Type SW2IVD-GPPQ GPP Software package OPERATING MANUAL (Q6TEL) Describes Q6TEL system configuration, operating methods, etc. (Included with product)	IB-66777 (13J924)

1 Using This Manual

1.1 Using This Manual

This manual is for users of MELSEC-QnA series PCs. It describes the specifications and functions of Q2ASCPU, Q2ASCPU-S1, Q2ASHCPU and Q2ASHCPU-S1 (hereinafter referred to as Q2ASCPU); specifications of other modules; and maintenance necessary to operate the system smoothly.

The manual is comprised of the following major chapters:

1) Chapters 2 to 3
   Describes Q2ASCPU overview and system configurations.
   Q2ASCPU features, modules used for system construction and precautions are described.

2) Chapters 4 to 15
   Describes Q2ASCPU specifications and functions.
   Various Q2ASCPU functions are described for effective use of Q2ASCPU.

3) Chapters 16 to 18
   Describes specifications and handling of modules (power supply, base, etc.) other than CPU module.
   Describes handling of the power-supply module, base, memory cards, and others necessary for the use of Q2ASCPU.

4) Chapters 19 to 21
   Describes Q2ASCPU installation and general-maintenance issues such as daily inspections and problem countermeasures.
   To help operate the Q2ASCPU smoothly it describes installation, daily inspections, and countermeasures in case of problems occurrences.

Remarks

This manual does not cover MELSECNET(II) Data Link, MELSECNET/B Data Link, MELSECNET/10 Network and SFC functions.
For each of these, refer to the following:

- MELSECNET(II) and MELSECNET/B Data Link
  Type MELSECNET and MELSECNET/B Data Link System Reference Manual

- MELSECNET/10 Network
  Type QnA/Q4AR MELSECNET/10 Network System Reference Manual

- SFC functions
  QCPU(Q mode)/QnACPU PROGRAMMING MANUAL (SFC)

1.2 Abbreviations and General Names Used in This Manual

The following abbreviations and general names are used in the manual:

(1) Q2ASCPU

Abbreviation for Q2ASCPU, Q2ASCPU-S1, Q2ASHCPU and Q2ASHCPU-S1-type CPU modules.

(2) Network module

Abbreviation for A1SJ71QLP21, A1SJ71QBR11 type MELSECNET/10 network modules.

(3) Ethernet module

Abbreviation for A1SJ71QE71N-B2, A1SJ71QE71N-B5T type Ethernet interface modules.

(4) Serial communication module

Abbreviation for A1SJ71QC24(N), A1SJ71QC24(N)-R2 type Serial communication modules.

(5) CC-Link

Abbreviation for Control & Communication Link.

(6) GPP function software

Abbreviation for SW□IVD/GPPQ type GPP function software package.
Abbreviation for GX Developer.

(7) DOS/V personal computer

Abbreviation for IBM PC/AT or 100% compatible.

(8) Peripheral devices for GPP functions

General name for a peripheral device such as a DOS/V personal computer used to operate GPP function software.

(9) Q6PU

Q6PU-type programming module.

(10) Peripheral device

General name for a peripheral device such as a DOS/V personal computer and Q6PU, which are connected to Q2ASCPU to operate it.

(11) Built-in RAM

The Q2ASCPU built-in RAM memory to store the sequence program

(12) Memory card

Abbreviation for Q1MEM-□□□-type memory cards

(13) ACPU

General name for a MELSEC-A series PC CPU.

2 Overview

2.1 Features

Q2ASCPU has the following new features:

(1) Increased memory capacity

(a) The program capacity of Q2AS(H)CPU-S1 has been increased to 60k steps (28k steps for Q2AS(H)CPU). Up to 60k steps can now be executed as one program.

(b) The device memory capacity has been increased to 29k words. Further, the user can now freely change the number of points for each device. For example, internal relay M has 8k points by default, but it can be expanded to 32k points.

(c) One memory card with a maximum 2 Mbytes memory can be inserted.
The memory card is used to store program codes, comments, statements, and file registers. (Q2ASCPU can be executed without a memory card, since the program can be stored in the CPU main module.)

(2) Enabled high-speed processing

(a) Speeded up operation processing of basic and application instruction.

	A2USCPU(S1)		Q2ASHCPU(S1)
Basic instructions:	0.2 μs	→	0.075 μs
Application instructions:	1.2 μs	→	0.225 μs

(b) The access time of extension data memory (file register R) is now equivalent to the access time of the Q2ASCPU built-in memory device (data register D and link register W)

(c) The buffer memory read/write time of the QnA dedicated special-function module (serial communication module) is approximately six times faster than AnUCPU. (The existing ACPU special function module has an equivalent processing speed AnUCPU.)

(d) The added high-speed access base module (A1S38HB) increases access processing speed by special function modules such as network modules and serial communication modules that handle vast amounts of data. By simply loading the special-function module rate the high-speed access base module, the Q2ASCPU can access special-function modules using higher processing speed.

(3) Enabled program-execution type selection for optimal control

The program-execution type can be selected from the following four types:

(a) Initial-execution type
Type of program to be executed only once when Q2ASCPU is set to RUN.

(b) Scan-execution type
Type of program executed continuously while the Q2ASCPU is running.
This is equivalent to the program from step 0 to end. Subroutine programs and an interrupt program can also be created in this program.

(c) Low-speed execution type
Type of program executed during the extra time of the constant-scan time (process to standardize scan time by specifying the program-execution time ahead of time).

(d) Standby type
Type of program composed from subroutine and interrupt programs only.

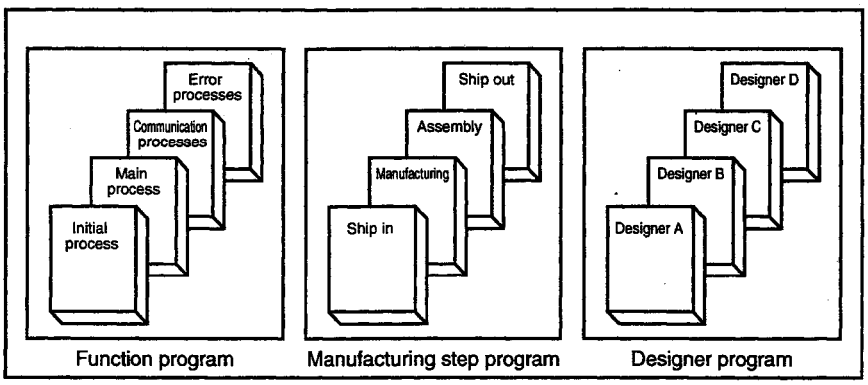
(4) Equipped with the SFC language MELSAP3.

SFC programming on MELSAP3 is much easier with the addition of step attributes and SFC control instructions.

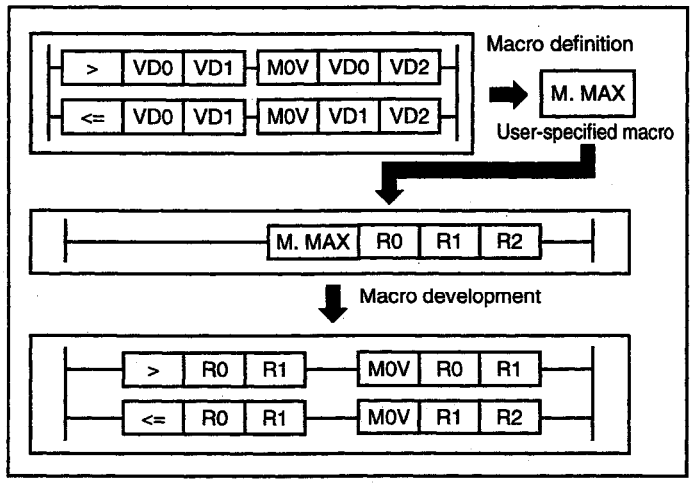
(5) Provides a more productive software-development environment.

(a) Enables structured programming through program component files.

Programs, which used to be one continuous module, can now be controlled as multiple file structures. Program design and control can now be divided up among multiple designers according to functions, processes or designer types.



(b) Users can create and use macro instructions as needed. Programming is standardized and simplified.



(c) Enabled wide variety of device handling

1) Enabled bit operation of word device
2) Can use differential contact point.
3) Program can directly access special function modules buffer memory as a device.
4) Program can directly access network module's data link as a device.

(d) Improved GPP function program editing operations
1) Up to four programs and data can be edited simultaneously.
Cut-and-paste operations for programs and data are possible among editing targets.
2) Circuits can be edited while displaying comments.
3) User-friendly operation is possible with pull-down menus and dialogue boxes.

(e) Improved debugging function for the system start-up
1) Enabled monitor circuit-correction steps
2) Enabled inspect on cause for coil on/off behavior
3) Enabled and optimized debugging environment by specifying monitoring timing through step number and device status
4) Enabled index-qualified device monitoring

(f) Improved GPP function document-creation function
1) The comment area is expanded to 32 characters; even more detailed comments can be attached.
2) Comments can be placed in all devices.
3) Statements and notes can be controlled as an integrated part of the program; program modification and carry over are easier.
4) Printout data can be stored in a file; printout time is shortened.

(g) The added powerful support software package to assist program creation
1) Data-conversion package
Converts comments and device data created by commercially available spreadsheet software and general-purpose text editors into GPPQ function files.
The GPP function files can also be converted into text or spreadsheet data.
2) Macro library package
Provides a macro library package for basic programs to access special-function modules as well as standard fault-detection and alarm-processing programs.
3) Ladder sequence generating & combining package
Combines multiple programs into one.
Enabled automatic assignment of program devices without duplication.
4) CAD interface package
This package can treat ladder circuits, instruction lists, comment data and SFC diagrams as data in order to communicate with CAD.

2.2 Q2ASCPU Additional Functions

New instructions for the functions and special function modules are added to Q2ASCPU.

[Additional functions]

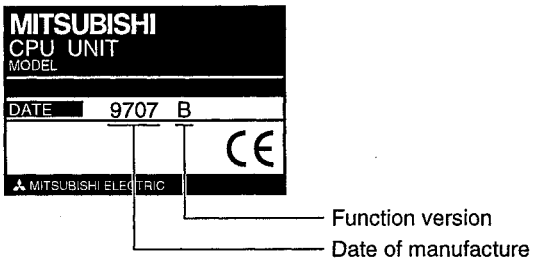
- Enhanced local device ..... See Section 2.2.1 (1).
  Monitoring and testing of local devices..... See Section 8.2.2.
  Use of the local device stored with the
  subroutine/interrupt program ..... See Appendix 7.
- CC-Link automatic refresh setting..... See Sections 2.2.1 (2) and 7.2
- MELSECNET/10 relay communication from Ethernet module
  (Network relay) ..... See Section 2.2.1 (3), Appendix (8).
- Additional A1SJ71QC24N compatible command . Section 2.2.1 (4).

[Additional instructions for the special function module]

The following are the additional instructions introduced in Q2ASCPU function version "B."

- A1SD61QBT11 control instruction ..... 13 instructions
- A1SD75 control instruction..... 19 instructions
- A1SJ71ID□-R4 control instruction ..... 12 instructions
- A1SJ71QE71 control instruction ..... 10 instructions

Above additional functions and instructions for the special function module may be used by Q2ASCPUs with function version "B" recorded in the DATE column on the rated plate.
In other words, when using the additional functions and instructions for the special function module, be sure to check that the function version "B" is recorded on the Q2ASCPU rated plate.
You may skip this section and subsequent explanation on additional functions if you're using a Q2ASCPU without the function version "B" recorded on the plate.



When using the above Q2ASCPU's additional functions and the instructions for the special function module, the GPP function model and function version or version of the applicable special function module need to be matched (See Table 2.1).

Table 2.1 Combination list of Q2ASCPU and function version/version of the special function module

Module/package name		Q2ASCPU	SW0IVD/ NX-GPPQ SW1IVD/ NX-GPPQ	SW2IVD/ NX-GPPQ	A1SJ71QE71 (B2), (B5)	A1SD75P-S3	A1SJ71 ID□-R4	A1SJ61QBT11	A1SJ71 QC24(N) (R2)
Condition	Function version	9707B and later	—	—	9707B and later	—	¤	9707B and later	—
	Version	—	No restrictions	No restrictions	—	No restrictions	BC and later	¤	No restrictions
Local device monitor test		○	×	○	—	—	—	—	—
Subroutine/interrupt program local device switch		○	—		—	—	—	—	—
CC-Link automatic refresh function		○	×	○	—	—	—	○	—
A1SJ61Q8T11 control instruction		○	○		—	—	—	○	—
MELSECNET/10 relay communication from Ethernet		○	×	△	○	—	—	—	—
A1SJ71QE71 control instruction		○	○		○	—	—	—	—
A1SD75 control instruction		○	○	○	—	○	—	—	—
ID interface module instruction		○	○		—	—	○	—	—
A1SJ71QC24N command support		○	—		—	—	—	—	○

Remarks

1) ○, —, △, and × marks in Table 2.1 denote the following explanations.
   ○ : Required for using functions and instructions.
   — : Not necessary for functions and instructions.
   △ : Required when accessing the Q2ASCPU of other stations via Ethernet from a peripheral device.
   × : Unusable function on peripheral device.
2) GX Developer support functions provided by the function version "B."

2.2.1 Summary of Additional Functions

This section describes an overview of the additional functions.

(1) Enhanced local device support

(a) The device set as a local device by "parameter device setting" function can now be monitored and tested from a peripheral device.
With this function, debugging may be performed while verifying the contents of a local device used by the program being monitored by a peripheral device.

(b) The local devices in a file where a subroutine or interrupt program is stored, can be used for executing subroutine/interrupt programs.
With this function, computation can be performed using the local devices stored in the file in which a subroutine or interrupt program is stored.

(c) When monitoring or testing a local device, the GPP function software packages listed below are required.

- IBM PC/AT or 100% compatibles : SW2IVD-GPPQ type GPP function software package.
  GX Developer.

(2) CC-Link automatic refresh setting

(a) By setting the CC-Link automatic refresh data using the peripheral functions, the set data allows for an automatic cyclic communication with other stations connected to the CC-Link.

- Remote I/O station (ON/OFF data communication)
- Remote device station (ON/OFF data + word data communication)
- Intelligent device station (ON/OFF data + word data communication)
- Local station/master station (ON/OFF data + word data communication)

Setting the CC-Link automatic refresh enables the use of FROM/TO instructions, which in turn allows for a direct communication with CC-Link's other stations without communicating with its master station.

(b) Automatic refresh setting can be conducted for up to 8 pieces of CC-Link modules per Q2ASCPU. For the ninth and subsequent CC-Link modules, communication can be performed using the FROM/TO instruction.

(c) The GPP function software packages listed below are required for the CC-Link automatic refresh setting.

- IBM PC/AT or 100% compatibles : SW2IVD-GPPQ type GPP function software package.
  GX Developer.

Also, the function version of modules for CC-Link mater station and local stations need to be "B" or later.

(3) Network relay from Ethernet module

(a) Data communication is allowed with Q2ASCPU of other stations via multiple Ethernet or MELSECNET/10 in the network system where Ethernet and MELSECNET/10 coexist.

(b) Network relay from an Ethernet module requires that with function version of "B" or later.

(4) A1SJ71QC24N support command enabled

(a) The following commands for A1SJ71QC24N are now supported.

- Multiple block batch read: command "0406"
- Multiple block batch write: command "1406"

(b) Multiple block "batch read"/"batch write" can be performed for A1SJ71QC24N (-R2, R4).
With A1SJ71QC24 (-R2, R4), multiple block "batch read"/"batch write" cannot be performed.
Refer to the manual below for multiple block "batch read"/"batch write" commands.

- Additional AJ71QC24N (-R2, R4) support instruction manual.

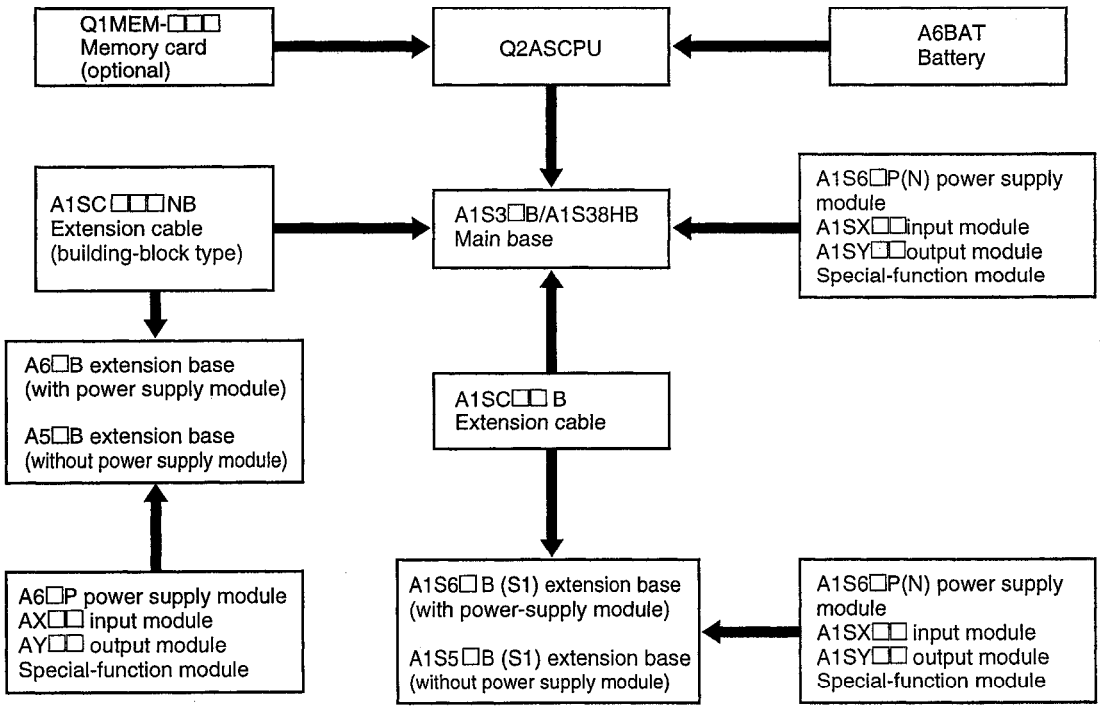
3 System Configuration

This section describes the system configuration, system precaution and devices as related to the Q2ASCPU.

3.1 System Configuration

Describes the system equipment and peripheral device configuration related to the Q2ASCPU independent system.

3.1.1 Independent System Machine Configuration

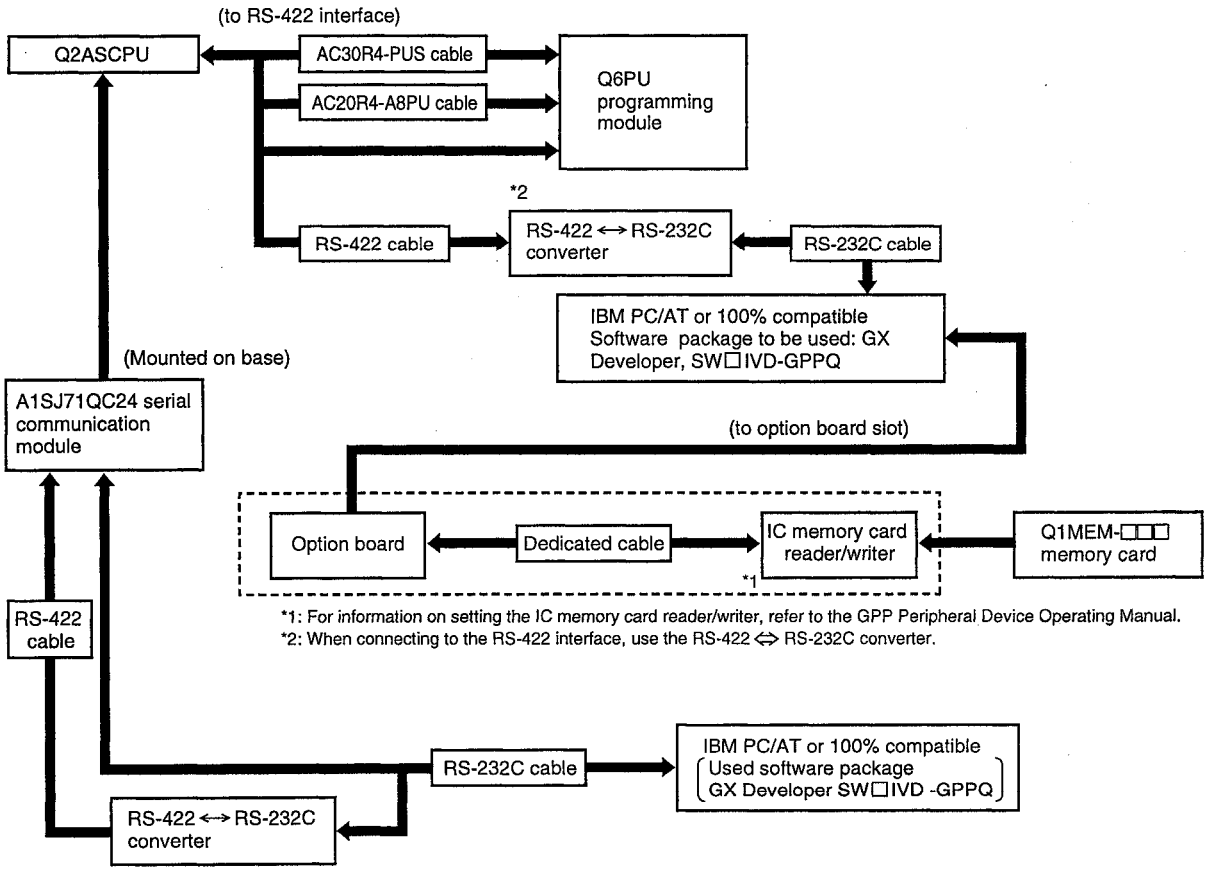


POINTS

*1 One memory card can be mounted as needed.
SRAM and E^2PROM memory cards can be mounted in the CPU module to enable read/write operations. When a flash memory card is installed to the CPU module, only the file-read operation is possible. Use a peripheral device to write files. (Refer to Section 3.1.2.)

*2 When using A1S5□B(S1) and A5□B extension base module, pay special attention to the power capacity of the main base module. I/O modules and special-function modules having substantial internal power consumption should be attached to the A1S6□B(S1) and A6□B extension bases. (For details, refer to Sections 16.1 and 17.3.)

3.1.2 Q2ASCPU Peripheral Device Configurations

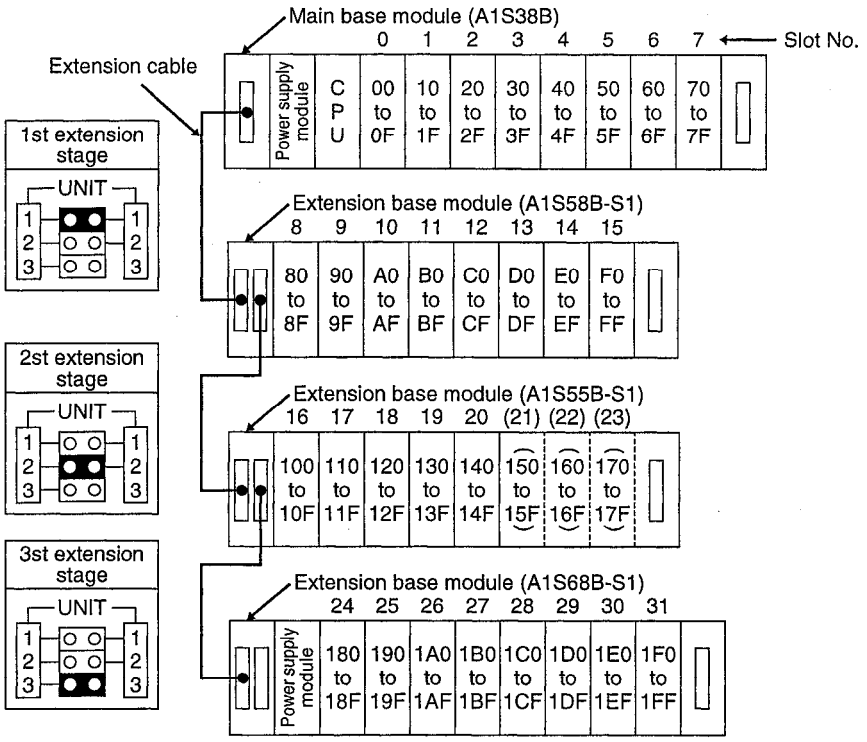

(to RS-422 interface)
Q2ASCPU
AC30R4-PUS cable
AC20R4-A8PU cable
Q6PU programming module
*2
RS-422 cable
RS-422 ↔ RS-232C converter
RS-232C cable
IBM PC/AT or 100% compatible Software package to be used: GX Developer, SW□IVD-GPPQ
(Mounted on base)
A1SJ71QC24 serial communication module
(to option board slot)
Option board
Dedicated cable
IC memory card reader/writer
*1
Q1MEM-□□□ memory card
RS-422 cable
*1: For information on setting the IC memory card reader/writer, refer to the GPP Peripheral Device Operating Manual.
*2: When connecting to the RS-422 interface, use the RS-422 ⇔ RS-232C converter.
RS-232C cable
IBM PC/AT or 100% compatible
(Used software package
GX Developer SW□IVD -GPPQ)
RS-422 ↔ RS-232C converter

Remarks

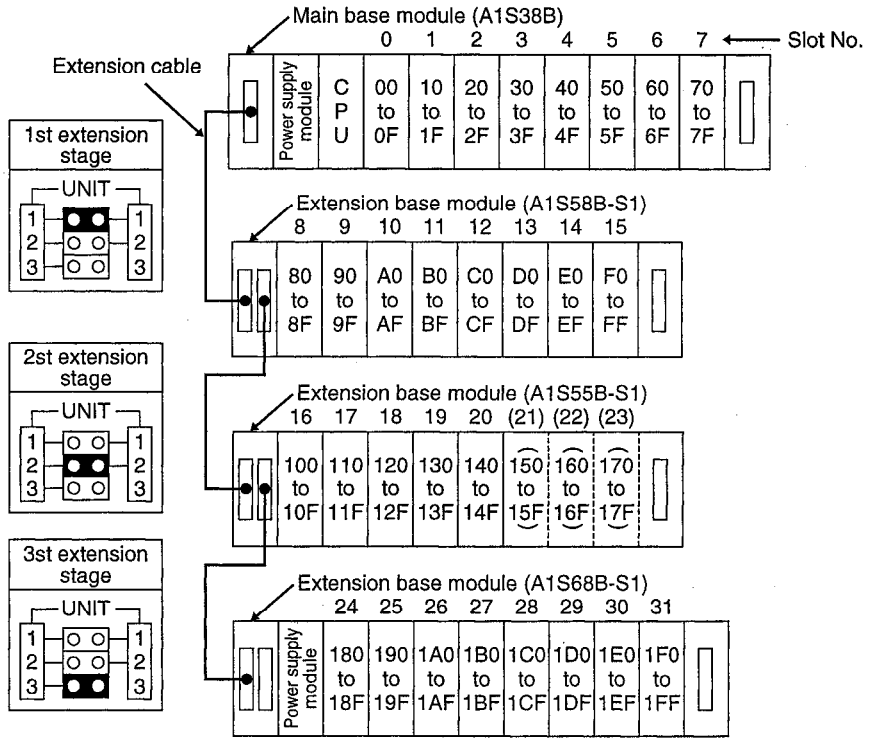
1) For details of system configurations for individual peripheral devices, refer to respective operating manuals.
2) The ACPU peripheral device can be connected to the Q2ASCPU only when accessing the ACPU of other stations through MELSECNET/10 network and MELSECNET data link. (The Q2ASCPU cannot be accessed, however.) In this case, SW1 of the system setup switch 2 on the CPU main module should be set to ON.

3.2 System Configuration Overview

(a) Q2ASCPU and Q2ASHCPU system

System configuration	 * The above figure shows the configuration when 16-input/output modules are loaded to each slot.
Maximum Number of Extension Stages	Three Extension Stages
Maximum number of input/output points	512 points
Main base modules	A1S32B, A1S33B, A1S35B, A1S38B, A1S38HB
Extension base modules	A1S52B(S1), A1S55B(S1), A1S58B(S1), A1S65B(S1), A1S68B(S1), A52B, AB55B, A58B, A62B, A65B, A68B
Extension cables	A1SC01B, A1SC03B, A1SC07B, A1SC12B, A1SC30B, A1SC60B, AC06B, AC12B, AC30B, A1SC05NB, A1SC07NB, A1SC30NB, A1SC50NB
Notes	(1) When extension base A1S52B(S1), A1S55B(S1), A1S58B(S1), and A52B, A55B, A58B are used, 5VDC is supplied from the main base's power supply module. Refer to Section 17.3 for the appropriate module to use. (2) Do not use extension cable longer than 6 m (19.68 ft.). (3) When using an extension cable, it should not be connected to or allowed to come close to the main circuit (high voltage and large current).
I/O number allocation	(1) I/O number is assigned first to the main base and then to the extension base. (2) In allocating I/O numbers to the main base and extension base, it is assumed that each of the bases has 8 slots. If 2/3/5 slot A1S32B/A1S33B/A1S35B is used for the main base, I/O numbers for the extension base are allocated after allowing for an additional 6/5/3 slots (96 points/80 points/48 points). (3) Allocate 16 points to a vacant slot. (4) Items (2) to (3) above can be modified through I/O allocation in the parameter. (Refer to Section 5.3.)

(b) Q2ASCPU-S1 and Q2ASHCPU-S1 system

System configuration	 * The above figure shows the configuration when 16-input/output modules are loaded to each slot.
Maximum Number of Extension Stages	Three Extension Stages
Maximum number of input/output points	1024 points
Main base modules	A1S32B, A1S33B, A1S35B, A1S38B, A1S38HB
Extension base modules	A1S52B(S1), A1S55B(S1), A1S58B(S1), A1S65B(S1), A1S68B(S1), A52B, A55B, A58B, A62B, A65B, A68B, A1SC30NB, A1SC50NB
Extension cables	A1SC01B, A1SC03B, A1SC07B, A1SC12B, A1SC30B, A1SC60B, AC06B, AC12B, AC30B, A1SC05NB, A1SC07NB, A1SC30NB, A1SC50NB
Notes	(1) When extension base A1S52B(S1), A1S55B(S1), A1S58B(S1), and A52B, A55B, A58B are used, 5VDC is supplied from the main base's power-supply module. Refer to Section 17.3 for the appropriate module to use. (2) Do not use extension cable longer than 6 m (19.68 ft.). (3) When using an extension cable, it should not be connected to or allowed to come close to the main circuit (high voltage and large current).
I/O number allocation	(1) I/O number is assigned first to the main base and then to the extension base. (2) In allocating I/O numbers to the main base and extension base, it is assumed that each of the bases has 8 slots. If 2/3/5 slot A1S32B/A1S33B/A1S35B is used for the main base, I/O numbers for the extension base are allocated after allowing for an additional 6/5/3 slots (96 points/80 points/48 points). (3) Allocate 16 points to a vacant slot. (4) Items (2) to (3) above can be modified through I/O allocation. (Refer to Section 5.3.)

3.3 List of System Equipment

3.3.1 List of System Equipment

The Q2ASCPU system can use the following modules and system equipment of peripheral device.

(1) QnA Modules

Name	Model	QnA only	QnA/A share	Description	Occupied points (I/O allocation module type)	Current consumption		Remark
						5VDC (A)	24VDC (A)	
CPU module	Q2ASCPU	—	—	512 I/O points, built-in RAM 28k step	—	0.3	—	Memory card purchased separately Includes memory card current consumption
	Q2ASHCPU			512 I/O points, built-in RAM 28k step		0.7	—	
	Q2ASCPU-S1			1024 I/O points, built-in RAM 60k step		0.3	—	
	Q2ASHCPU-S1			1024 I/O points, built-in RAM 60k step		0.7	—	
Memory card	Q1MEM-64S	○	×	SRAM 64 kb	—	—	—	
	Q1MEM-128S			SRAM 128 kb				
	Q1MEM-256S			SRAM 256 kb				
	Q1MEM-512S			SRAM 512 kb				
	Q1MEM-1MS			SRAM 1 Mb				
	Q1MEM-2MS			SRAM 2 Mb				
	Q1MEM-64SE			SRAM 32 Kb, E^2PROM 32 kb				
	Q1MEM-128SE			SRAM 64 Kb, E^2PROM 64 kb				
	Q1MEM-256SE			SRAM 128 Kb, E^2PROM 128 kb				
	Q1MEM-512SE			SRAM 256 Kb, E^2PROM 256 kb				
	Q1MEM-1MSE			SRAM 512 Kb, E^2PROM 512 kb				
	Q1MEM-256SF			SRAM 128 Kb, flash memory 128 kb				
	Q1MEM-512SF			SRAM 256 Kb, flash memory 256 kb				
	Q1MEM-1MSF			SRAM 512 Kb, flash memory 512 kb				
	Q1MEM-2MSF			SRAM 1 Mb, flash memory 1 Mb				

Remarks

The Q-dedicated module can be used on Q2ASCPU only.
The Q/A-common module can be used on either Q2ASCPU or ACPU. (However, some restrictions may apply for use with Q2ASCPU.)

Name	Model	QnA only	QnA/A share	Description		Occupied points (I/O allocation module type)	Current consumption 5VDC (A)	Current consumption 24VDC (A)	Remark
Power supply module	A1S61PN	×	○	5VDC, 5A	Input 100VAC to 200VAC	—	—	—	For power supply slots of main base and extension base
	A1S62PN			5VDC, 3A/24VDC, 0.6A					
	A1S63P			5VDC, 5A	Input 24VDC				
Input module	A1SX10	×	○	16 points 100VAC input module		16[Input 16 points]	0.05	—	
	ASX10EU								
	A1SX20			16 points 200VAC input module		16[Input 16 points]	0.05	—	
	A1SX20EU								
	A1SX30			16 points 12/24VDC, 12/24VAC input module		16[Input 16 points]	0.05	—	
	A1SX40			16 points 12/24VDC input module		16[Input 16 points]	0.05	—	
	A1SX40-S1			16 points 24VDC input module		16[Input 16 points]	0.05	—	
	A1SX40-S2			16 points 24VDC input module		16[Input 16 points]	0.05	—	
	A1SX41			32 points 24VDC input module		32[Input 32 points]	0.08	—	
	A1SX41-S1			32 points 24VDC input module		32[Input 32 points]	0.12	—	
	A1SX41-S2			32 points 24VDC input module		32[Input 32 points]	0.08	—	
	A1SX42			64 points 12/24VDC input module		64[Input 64 points]	0.09	—	
	A1SX42-S1			64 points 24VDC input module		64[Input 64 points]	0.16	—	
	A1SX42-S2			64 points 24VDC input module		64[Input 64 points]	0.09	—	
	A1SX71			32 points 5/12VDC input module		32[Input 32 points]	0.075	—	
	A1SX80			16 points 12/24VDC Sink/source input module		16[Input 16 points]	0.05	—	
	A1SX80-S1			16 points 24VDC Sink/source input module		16[Input 16 points]	0.05	—	
	A1SX80-S2			16 points 24VDC Sink/source input module		16[Input 16 points]	0.05	—	
	A1SX81			32 points 12/24VDC Sink/source input module		32[Input 32 points]	0.08	—	
	A1SX81-S2			32 points 24VDC Sink/source input module		32[Input 32 points]	0.08	—	
	A1SX82-S1			64 points 24VDC input module		64 [Input 64 points]	0.16	—	
Output module	A1SY10	×	○	16 points Relay contact output module (2A)		16[Output 16 points]	0.12	0.09	
	A1SY10EU								
	A1SY14EU								
	A1SY18A			8 points Relay contact output module (2A) for independent contact		16[Output 16 points]	0.24	0.075	
	A1SY18AEU								
	A1SY22			16 points triac output module (0.6A)		16[Output 16 points]	0.27	(200VAC) 0.004	
	A1SY28A			8 points triac output module (1A) all points independent		16[Output 16 points]	0.11	—	
	A1SY28EU								
	A1SY40			16 points 12/24VDC transistor output module (0.1A) sink type		16[Output 16 points]	0.27	0.016	
	A1SY41			32 points 12/24VDC transistor output module (0.1A) sink type		32[Output 32 points]	0.50	0.016	
	A1SY42			64 points 12/24VDC transistor output module (0.1A) sink type		64[Output 64 points]	0.93	0.016	
	A1SY50			16 points 12/24VDC transistor output module (0.5A) sink type		16[Output 16 points]	0.12	0.12	
	A1SY60			16 points 24VDC transistor output module (2A) sink type		16[Output 16 points]	0.12	0.015	
	A1SY60E			16 points 12VDC transistor output module (1A) source type		16[Output 16 points]	0.20	0.01	
	A1SY68A			8 points 5/12/24/48VDC transistor output module sink/source type all points independent		16[Output 16 points]	0.13	—	
	A1SY71			32 points 5/12VDC transistor output module (0.016A) sink type		32[Output 32 points]	0.40	0.15	
	A1SY80			16 points 12/24VDC transistor output module (0.8A) source type		16[Output 16 points]	0.12	0.04	
	A1SY81			32 points 12/24VDC transistor output module (0.1A) source type		32[Output 32 points]	0.50	0.016	
	A1SY81EP			32 points 12/24VDC transistor output module (0.1A/0.05A) source type		32[Output 32 points]	0.50	0.016	
	A1SY82			64 points 12/24VDC transistor output module (0.1A) source type		64 [Output 64 points]	0.93	0.016	

Name	Model	QnA only	QnA/A share	Description	Occupied points (I/O allocation module type)	Current consumption		Remark
						5VDC (A)	24VDC (A)	
I/O module	A1SH42	×	○	32 points 12/24VDC input module 32 points 12/24VDC transistor output module (0.1A) sink type	32[Output 32 points]	0.50	0.008	
	A1SH42-S1			32 points 24VDC input module 32 points 24VDC transistor output module (0.1A) sink type	32 [Output 32 points]	0.50	0.008	
	A1SX48Y18			8 points 24VDC input module 8 points Relay contact output module	16[Output 16 points]	0.085	0.045	
	A1SX48Y58			8 points 24VDC input module 8 points 12/24VDC transistor output module	16[Output 16 points]	0.06	0.06	
Dynamic input module	A1S42X	×	○	16/32/48/64 points 12/24VDC dynamic input module	Specified points [Input {Specified points}]	0.08	—	
Dynamic output module	A1S42Y	×	○	16/32/48/64 points 12/24VDC dynamic output module	Specified points [Output {Specified points}]	0.10	0.008	
Blank cover	A1SG60	×	○	Dust cover for unused slot	16[blank]	—	—	
Dummy module	A1SG62	×	○	16 point, 32 point, 48 point, 64 point selectable module	Specified points [Input {Specified points}]	—	—	
40 pin connector	A6CON1	×	○	Soldering type (straight out type)	—	—	—	
	A6CON2			Crimp-contact type (straight out type)				
	A6CON3			Pressure-displacement type (flat cable type)				
	A6CON4			Soldering type (straight/diagonal out type)				
37 pin D subconnector	A6CON1E	×	○	Soldering type (straight out type)	—	—	—	
	A6CON2E			Crimp-contact type (straight out type)				
	A6CON3E			Pressure-displacement type (flat cable type)				
Pulse catch module	A1SP60	×	○	Pulse input module with short pulse duration (minimum pulse duration:0.5 ms) Input 16 points	16[Output 16 points]	0.055	—	
Analog timer module	A1ST60	×	○	Allows for different set timer value (0.1 to 1.0 s, 1 to 10 s, 10 to 60 s, 60 to 600 s) depending on volume Analog timer 8 points	16[Output 16 points]	0.055	—	
Interrupt module	A1SI61	×	○	For interrupt program execution Interrupt module (Interrupt input:16 points)	32[Special 32 points]	0.057	—	
PLC simple monitor module	A1SS91	×	○	PLC external failsafe circuit simplifying module	16 [Output 16 points]	0.8	—	
High-speed counter module	A1SD61	×	○	32 bit signed binary. 50 KPPS, 1 channel	32[Special 32 points]	0.35	—	
	A1SD62			24 bit binary Sink output type 100 KPPS, 2 channels	32[Special 32 points]	0.1	—	
	A1SD62E			24 bit binary Source output type 100 KPPS, 2 channels	32[Special 32 points]	0.1	—	
	A1SD62D			24 bit binary Differential input type 200 KPPS, 2 channels	32[Special 32 points]	0.25	—	
	A1SD62D-S1			24-bit binary 200KBPS, 2 channels	32[Special 32 points]	0.25	—	

Name	Model	QnA only	QnA/A share	Description	Occupied points (I/O allocation module type)	Current consumption 5VDC (A)	Current consumption 24VDC (A)	Remark
Temperature adjustment module	A1S62TCTT-S2	×	○	Transistor output, thermocouple input 2 channels/module PID control : ON/OFF pulse	32[Special 32 points]	0.28	—	
	A1S62TCTTBW -S2	×	○	Transistor output, thermocouple input 2 channels/module PID control : ON/OFF pulse, heater wire breakage detection function			—	
	A1S62TCRT-S2	×	○	Transistor output, platinum temperature-measuring resistor input 2 channels/module PID control : ON/OFF pulse	32[Special 32 points]	0.28	—	
	A1S62TCRTBW -S2	×	○	Transistor output, platinum temperature-measuring resistor input 2 channels/module PID control : ON/OFF pulse, heater wire breakage detection function			—	
	A1S64TCTT-S1	×	○	Transistor output, thermocouple input 4 channels/module PID control : ON/OFF pulse or 2- position control	32[Special 32 points]	0.42	—	
	A1S64TCTTBW -S1	×	○	Transistor output, thermocouple input 4 channels/module PID control : ON/OFF pulse or 2-position control Heater wire breakage detection function			—	
	A1S64TCRT-S1	×	○	Transistor output, thermocouple input 4 channels/module PID control : ON/OFF pulse or 2-position control	32[Special 32 points]	0.42	—	
	A1S64TCRTBW -S1	×	○	Transistor output, thermocouple input 4 channels/module PID control : ON/OFF pulse or 2-position control Heater wire breakage detection function			—	
Analog-digital conversion module	A1S64AD	×	○	0 to 10 V DC / DC0 to 20 mA Analog input 4 channels	32[Special 32 points]	0.4	—	
	A1S68AD			0 to 10 V DC / DC0 to 20 mA Analog input 8 channels	32[Special 32 points]	0.4	—	
Temperature-digital conversion module	A1S62RD3	×	○	For connecting Pt 100 (3-wire type) Temperature input 2 channels	32[Special 32 points]	0.54	—	
	A1S62RD4			For connecting Pt 100 (4-wire type) Temperature input 2 channels	32[Special 32 points]	0.44	—	
Digital-analog conversion module	A1S62DA	×	○	DC4 to 20 mA / 0 to 10 V DC Analog output 2 channels	32[Special 32 points]	0.8	—	
	A1S68DAV			0 to 10 V DC, Analog output 8 channels	32[Special 32 points]	0.65	—	
	A1S68DAI			DC4 to 20 mA, Analog output 8 channels	32[Special 32 points]	0.85	—	
Computer link module	A1SJ71QC24N	○	×	Computer link function, RS-232C 1 channel, RS-422/485 1 channel	32[Special 32 points]	0.35	—	
	A1SJ71QC24N -R2			Computer link function RS-232C 2 channel	32[Special 32 points]	0.3	—	
	A1SJ71UC24-R2	×	○	Computer link function RC-232C 1 channel	32[Special 32 points]	0.1	—	Accessible in Refer to Section 3.3.2.
	A1SJ71UC24-PRF			Computer link & printer function RS-232C 1 channel	32[Special 32 points]	0.1	—	
	A1SJ71UC24-R4			Computer link function RS-422 / RS-485 1 channel	32[Special 32 points]	0.1	—	
Data transfer interface module	A1SJ71CM0-S3	×	○	Computer link function (dedicated protocol, non-procedural mode) built-in modem, number of connected lines: 1	32[Special 32 points]	0.26	—	Refer to Section 3.3.2 for the access range.
Positioning module	A1SD70	×	○	For one axis positioning control, speed control, speed-positioning, Analog output voltage (0 to 10 V DC)	32[Special 32 points]	0.3	—	
	A1SD71-S2			For positioning control, speed control, speed-positioning control, Pulse chain output, Two axes (independent, 2 axes synch, linear interpolation)	48[Special 48 points]	0.8	—	
	A1SD71-S7			For positioning control, selectable manual - pulsar output speed, Pulse chain output, Two axes (independent, 2 axes synch, linear interpolation)	48[Special 48 points]	0.8	—	

Name	Model	QnA only	QnA/A share	Description	Occupied points (I/O allocation module type)	Current consumption 5VDC (A)	Current consumption 24VDC (A)	Remark
Positioning module	A1SD75P1-S3	×	○	For positioning control, Pulse chain output, One axis	32[Special 32 points]	0.7	—	* When Differential driver is connected: 0.78
	A1SD75P2-S3			Used for positioning control Pulse output, 2 axes (independent, dual-axis simultaneous, linear interpolation, circular interpolation)				
	A1SD75P3-S3			Used for positioning control Pulse output, 3 axes (independent, triple-axis simultaneous, dual-axis linear interpolation, dual-axis circular interpolation)		0.7 *		
	A1SD75M1			Used for positioning control Digital output. For MR-H-B/MR-J-B/MR-J2-B. 1 axis.		0.7		
	A1SD75M2			Used for positioning control Digital output. For MR-H-B/MR-J-B/MR-J2-B. 2 axes. (independent, dual-axis simultaneous, linear interpolation, circular interpolation)				
	A1SD75M3			Used for positioning control Digital output. For MR-H-B/MR-J-B/MR-J2-B. 3 axes. (independent, triple-axis simultaneous, dual-axis linear interpolation, dual-axis circular interpolation)				
Position detection module	A1S62LS	×	○	Absolute detection system Resolution: 1 resolver revolution = 4096-part split Response speed: Within 2 ms	48:First half ; Special 32 points 32:Second half; Vacant 16 points	0.55	—	
Analog I/O module	A1S63ADA	×	○	Analog input, Two channels Controllable simplified loop, Analog output. 1 channel	32[Special 32 points]	0.8	—	
	A1S66ADA			Analog input : 4 channels I/O signal system Analog output : 2 channels	64 [Output 64-point]	0.16	—	
Thermocouple input module	A1S68TD	×	○	For thermocouple connection Temperature input 8 channels	32[Special 32 points]	0.32	—	
MELSECNET(II) data link module	A1SJ71AP21	×	○	For MELSECNET(II) data link system master station and local station (for S1 type optical fiber cable)	32[Special 32 points]	0.33	—	Accessible in the A2ACPU device range
	A1SJ71AP21-S3			For MELSECNET(II) data link system master station and local station (for G1-type optical fiber cable)	32[Special 32 points]	0.33	—	
	A1SJ71AR21			For MELSECNET(II) data link system master station and local station (for coaxial cable)	32[Special 32 points]	0.8	—	
MELSECNET/B data link module	A1SJ71AT21B	×	○	For MELSECNET/B data link system master station and local station	32[Special 32 points]	0.66	—	
	A1SJ72T25B			For MELSECNET/B data link system remote I/O station	—	0.3	—	
MELSECNET/10 datalink module	A1SJ71QLP21	○	×	For control station , master station , and local station of the MELSECNET/10 data link module system (For SI type optical fiber cable, double loop)	32 [Special 32-point]	0.65	—	
	A1SJ71QBR11	○	×	For control station , master station, and local station of the MELSECNET/10 data link module system (For coaxial cable single bus)	32 [Special 32-point]	0.80	—	
CC-Link system master-local module	A1SJ61QBT11	○	×	For the master station of the CC-Link data link system (Dedicatedly for shielded twisted pair cable)	32 [Special 32-point]	0.40	—	
MELSECNET/MINI-S3 master module	A1SJ71PT32-S3	×	○	For MELSECNET/MINI-S3 master station, 64 stations maximum, Controls remote I/O with a total of 512 I/O points, and remote terminal (for optical fiber cable/twisted pair cable)	I/O Dedicated mode. 32 [special 32 points] Extension mode. 48 [special 48 points]	0.35	—	
	A1SJ71T32-S3			For MELSECNET/MINI-S3 master station, 64 stations maximum, Controls remote I/O with a total of 512 I/O points, and remote terminal (for twisted pair cable only)	I/O Dedicated mode. 32 [special 32 points] Extension mode. 48 [special 48 points]	0.3	—	
MELSEC - I/O LINK master module	A1SJ51T64	×	○	For MELSECNET - I/O master station, 16 stations maximum, Controls I/O LINK with a total of 128 I/O points, and remote I/O module	64[Special 64 points]	0.115	0.09	

Name	Model	QnA only	QnA/A share	Description	Occupied points (I/O allocation module type)	Current consumption 5VDC (A)	Current consumption 24VDC (A)	Remark
B/NET interface module	A1SJ71B62-S3	×	○	For B/NET transmission terminal control Up to 63 stations can be controlled per module.	32[Special 32 points]	0.08	—	
S-LINK system master module	A1SJ71SL92	×	○	For S-LINK device control Number of control I/O points: Max. 128 points	32[Special 32 points]	0.2	—	
JEMANET (JPCN-1) interface module	A1SJ71J92-S3	×	○	JEMANET (JPCN-1) master module	32[Special 32 points]	0.4	—	
	A1SJ72J95			JEMANET (JPCN-1) slave module	32[Special 32 points]	0.4	—	
DeviceNet interface module	A1SJ71DN91	×	○	Master module for DeviceNet, total I/O: 4096 points	32[Special 32 points]	0.24	—	
PROFIBUS-DP interface module	A1SJ71PB92D	×	○	Master module for PROFIBUS-DP, data that can be sent Normal service: 32 bytes Extended service: 244 bytes	32[Special 32 points]	0.56	—	
PROFIBUS-DP slave module	A1SJ71PB93D	×	○	Slave module for PROFIBUS-DP, total I/O data: 192 words	32[Special 32 points]	0.36	—	
PROFIBUS-FMS interface module	A1SJ71PB96F	×	○	Master/client/server module for PROFIBUS-FMS, total I/O data: 241 bytes	32[Special 32 points]	0.56	—	
AS-I interface module	A1SJ71AS92	×	○	Master module for AS-I, total I/O: 496 points	32[Special 32 points]	0.15	—	
MODBUC serial communication module	A1SJ71UC24-R4 -S2	×	○	Serial communication module for MODBUS, transmission speed: 300 to 19,200bps	32[Special 32 points]	1.4	—	Accessible in the Refer to Section 3.3.2.
Intelligent communication module	A1SD51S	×	○	Interpreter BASIC, Compiler BASIC. RS-232C 1 channel. RS-422/485 1 channel	32[Special 32 points]	0.4	—	Accessible in the Refer to Section 3.3.2.
Memory card interface module	AS1D59J-S2	×	○	Usable memory card JEIDA Ver. 4.0 compliant	32[Special 32 points]	0.05	—	Power consumption assumes connection of A1SD59J-MIF.
	AS1D59J-MIF							
Ethernet interface module	A1SJ71E71N -B2	×	○	For 10 BASE2 (Cheapernet)	32[Special 32 points]	0.64	—	Accessible in the Refer to Section 3.3.2.
	A1SJ71E71N -B5T			For 10 BASE5, 10 BASE-T (Ethernet)	32[Special 32 points]	0.42	—	
	A1SJ71QE71-B2	○	×	For 10Base2 (Cheapernet)	32[Special 32 points]	0.8	—	
	A1SJ71QE71-B5			For 10Base5 (Ethernet)	32[Special 32 points]	0.6	—	
ID interface module	A1SD35ID1	×	○	ID Interface module Connectable reader/writer unit: one	32[Special 32 points]	0.25	0.1	
	A1SD35ID2			ID Interface module Connectable reader/writer units: two	32[Special 32 points]	0.25	0.15	
Paging module	A1SD21-S1	×	○	Number of entries for calling control trigger device 2000 : maximum word number of 40 (Half-character) 1120 : maximum word number of 80 (Half-character) Number of entries for receiver number : 1000	32[Special 32 points]	0.14	—	
Simulation module	A6SIM-X64Y64	×	○	I/O simulation module for connection to the main base, Allows desk debugging without connecting I/O module to the base module.	64[Input 64 points] 64[Output 64 points]	0.3	—	Main base cannot be set while using the A1S38H
High-speed main base module	A1S38HB	○	×	May take 8 I/O modules	—	—	—	Extension connector attached to one on each side
Main- base module	A1S32B	×	○	May take 2 I/O modules				
	A1S33B			May take 3 I/O modules				
	A1S35B			May take 5 I/O modules				
	A1S38B			May take 8 I/O modules				

Name	Model	QnA only	QnA/A share	Description	Occupied points (I/O allocation module type)	Current consumption		Remark
						5VDC (A)	24VDC (A)	
Extension base module	A1S52B	×	○	May take 2 I/O modules	—	—	—	Does not take power supply module. (Power supplied from the main base module).
	A1S55B			May take 5 I/O modules				
	A1S58B			May take 8 I/O modules				
	A1S65B			May take 5 I/O modules	—	—	—	Requires a power supply module.
	A1S68B			May take 8 I/O modules				
Extension cable	A1SC01B			Flat cable, 55 mm (2.17 inch) long	—	—	—	For extension to right side
	A1SC03B			300 mm (11.8 inch) long	—	—	—	Extension base module connecting cable
	A1SC07B			700 mm (27.6 inch) long				
	A1SC12B			1200 mm (47.24 inch) long				
	A1SC30B			3000 mm (118.11 inch) long				
	A1SC60B			6000 mm (236.22 inch) long				
	A1SC05NB			450 mm (17.72 inch) long	—	—	—	Cable for A□N and A□A extension bases
	A1SC07NB			700 mm (27.6 inch) long				
	A1SC30NB			3000 mm (118.3 inch) long				
	A1SC50NB			5000 mm (197.1 inch) long				

Name	Model	QnA only	QnA/A share	Description	Applicable model
Battery	A6BAT	×	○	IC-RAM memory backup	Installed in the Q2AS(H)CPU(S1) main module
Connector/ terminal block conversion module	A6TBXY36	×	○	For sink type input module and sink type output module (standard type)	A1SX41(S1/S2), A1SX42(S1/S2), A1SX82-S1, A1SY41, A1SY42, A1SY82, A1SH42(S1)
	A6TBXY54			For sink type input module and sink type output module (2 wire type)	AX42(S1), AY42(S4), AH42, AJ35TC1-32D
	A6TBX70			For sink type input module (3 wire type)	A1SX41(S1/S2), A1SX42(S1/S2), A1SH42(S1), AX42(S1), AH42
	A6TBX36-E			For source type input module(standard type)	A1SX81(S1/S2), AX82
	A6TBY36-E			For source type output module(standard type)	A1SY81
	A6TBX54-E			For source type input module (2 wire type)	A1SX81(S1/S2), AX82
	A6TBY54-E			For source type output module (2 wire type)	A1SY81
	A6TBX70-E			For source type input module (3 wire type)	A1SX81(S1/S2), AX82
Cable for connector/ terminal block conversion module	AC05TB	×	○	0.5 m (1.64 ft.) long, for sink module	A6TBXY36 A6TBXY54 A6TBX70
	AC10TB			1 m (3.28 ft.) long, for sink module	
	AC20TB			2 m (6.56 ft.) long, for sink module	
	AC30TB			3 m (9.84 ft.) long, for sink module	
	AC50TB			5 m (16.4 ft.) long, for sink module	
	AC80TB			8 m (26.24 ft.) long, for sink module	
	AC100TB			10 m (32.8 ft.) long, for sink module	
	AC05TB-E			0.5 m (1.64 ft.) long, for source module	A6TBX36-E A6TBY36-E A6TBX54-E A6TBY54-E A6TBX70-E
	AC10TB-E			1 m (3.28 ft.) long, for source module	
	AC20TB-E			2 m (6.56 ft.) long, for source module	
	AC30TB-E			3 m (9.84 ft.) long, for source module	
	AC50TB-E			5 m (16.4 ft.) long, for source module	
Relay terminal module	A6TE2-16SRN	×	○	For sink type output module	A1SY41, A1SY42, A1SH42, AY42, AY42(S1/S3/S4), AH42
Cable for relay terminal module connection	AC06TE	×	○	0.6 m (1.97 ft.) long	A6TE2-16SR (N)
	AC10TE			1 m (3.28 ft.) long	
	AC30TE			3 m (9.84 ft.) long	
	AC50TE			5 m (16.4 ft.) long	
	AC100TE			10 m (32.8 ft.) long	
Terminal block cover for AnS I/O module and special module	A1STEC-S	×	○	Slim type terminal block cover for AnS I/O module and special module	All terminal block connector type modules
Pressure-displacement type terminal block adapter for AnS I/O module	A1S-TA32	×	○	Pressure-displacement type terminal block adapter for AnS 32 point I/O module	A1SX41(S1/S2), A1SX71, A1SY41, A1SY71

Remarks

I/O cables with connectors for I/O modules of 40-pin connector specifications (A1SX41, A1SX42, A1SY41, A1SY42, etc.) or 37-pin D-sub connector specificaions(A1SX81, A1SY81) are available. Consult the neareset Mitsubishi representative for the I/O cables with connectors.

(2) Peripheral device

Item	Model	Remarks
Programming module	Q6PU	Performs program write/read by connecting with the CPU main module via RS-422 cable (AC30R4-PUS, AC20R4-A8PU) (5 V DC 0.4 A)
Cable for RS-422	AC30R4-PUS	Connecting cable between the CPU main module and Q6PU. 3 m (9.84 ft.) long.
	AC20R4-A8PU	Connecting cable between the CPU main module and Q6PU. 2 m (6.56 ft.) long.

3.3.2 Precaution on System Configuration

This section describes hardware and software packages compatible with Q2ASCPU.

(1) Hardware

(a) Number of allowable units is limited depending on particular module.

Applicable module	For Q2ASCPU only	For ACPU	Remarks
I/O module	——	No limit	——
Special function module	——	No limit	——
Intelligent special function module	No limit	Total of 6 units	Including bus-connected GOT
Interrupt module	——	One unit only	——
Link module Ethernet module	Ethernet module for network, total of 4 units	For data link, total of 2 units	Ethernet module for network and data link, total of 4 units

Remarks

Further remarks on above modules.

1) I/O module
   Standard input module and output module
2) Special function module
   Special function module that can process FROM/TO instructions from Q2ASCPU (e.g., A1S64AD, A1S62DA, etc.)
3) Intelligent special function module
   Special function module that can not only processes FORM/TO instructions from Q2ASCPU, but also be able to access Q2ASCPU from the special function module (e.g., A1SJ71UC24, etc.)
4) Interrupt module
   Module that generates interruption to Q2ASCPU. (A1SI61)
5) Link module
   Special function module for MELSECNET II & /B data link as well as MELSECNET/10 network.
6) Ethernet module
   Ethernet interface module (A1SJ71QE71) dedicated to Q2ASCPU.

(b) The following special function modules cannot be used by Q2ASCPU:

- AJ71C23 (Host controller high-speed link module)
- AD57-S2 (A6MD controller module)
- AJ71C24 (Computer link module) ..... Products up until February, 1987
- (Products dated March, 1987 or later, or those with "H" (A3H compatible)displayed in date field, are Q2ASCPU compatible.)
- AD51 (Intelligent communication module) ..... Products up until March, 1987
- (Products dated April 1987 or later, or those with "H" (A3H compatible) in date field, are Q2ASCPU compatible.)
- A7GT-BUS (A77GOT-S5, A870GOT bus connection interface module)
  ..... Products up until January, 1996
- (Those products manufactured after February, 1996, or those marked as hardware version "C" in the DATE column, can be used.)
- AJ71LP21(G), AJ71BR11, AJ71LR21, A1SJ71LP21, A1SJ71BR11, A1SJ71LR21 (MELSECNET/10 network module)
- AJ72LP25, AJ72BR15 (MELSECNET/10 remote I/O station module)

(c) When using special function modules for Q2ASCPU, the following devices can be used according to each models:

Device \ Model	AD51(S3), AJ71C24-S3, AJ71P41 *1	AD51H(S3), AD51FD-S3, AJ71C23-S3, AJ71C24-S6/S8, AJ71UC24 *2	AJ71E71 *2	
			Q2AS(H)	Q2AS(H)-S1
I/O device (X / Y)	X/Y0 to 7FF	X/Y0 to 7FF	0 to 1FF	0 to 3FF
Internal relay (M, L, S) *3	M0 to 2047	M0 to 8191	M0 to 8191	
Link relay (B)	B0 to 3FF	B0 to FFF	B0 to FFF	
Timer (T)	T0 to 255	T0 to 2047	T0 to 2047	
Counter (C)	C0 to 255	C0 to 1023	C0 to 1023	
Data register (D)	D0 to 1023	D0 to 6143	D0 to 6143	
Link register (W)	W0 to 3FF	W0 to FFF	W0 to FFF	
Annunciator (F)	F0 to 255	F0 to 2047	F0 to 2047	

*1: Can access the A3HCPU-equivalent device range only.
Cannot perform read/write, etc. of file register data and program.
*2: Can access the AnACPU-equivalent device range only.
Cannot perform read/write, etc. of file register data and program.
*3: Acts as M device if L or S is specified. (Example: L10→M10)

(d) May attach Q2ASCPU to the main base module for A1S38HB type-high speed access in order to speed up the read/write process of link module buffer memory, special function buffer memory, and intelligent function buffer memory.

(e) The followings are methods to connect graphic operation terminal module to Q2ASCPU.

Model	Connection method	Accessible device range
A985GOT A975GOT A970GOT A960GOT A956GOT A956WGOT A951GOT	Direct CPU connection Computer link connection, CC-Link connection MELSECNET(II) connection MELSECNET/B connection MELSECNET/10 connection Bus connection	The entire Q2ASCPU device range is accessible. (For details, refer to SW□D5C-GOTR-PACK Operating Manual.)

(f) Accessible range of computer link module A1SJ71UC24 includes CPU module (host station) where A1SJ71UC24 is attached, and other stations on host station network. Can not access other stations on other networks under MELSECNET/10 network system routing function.
Accessible range of serial communication module AJ71QC24 includes host station, other stations on host station network, and other stations relayed through maximum of seven stations via routing function.

(g) When the intelligent communication module (AD51H-S3) accesses to Q2ASCPU/QnACPU on other stations in the network, only the control table-format 1 can be used.
The control table-format 2 cannot be used.
However, even with the control table-format 1, access to other networks cannot be performed.

(2) Software package

The following is Q2ASCPU system startup software packages for program creation:

Peripheral device for the GPP function	Software package for system startup
IBM PC/AT or 100% compatible	GX Developer SW□IVD-GPPQ

Set to the following PC CPU type using peripheral device.

PC CPU model	PC CPU type
Q2AS(H)CPU	Q2A
Q2AS(H)CPU-S1	Q2AS1

In addition to the above, following software packages can also be used.

- CAD interface package — SW0NX-CADQ, SW0IVD-CADQ
- Data conversion package — SW0NX-CNVQ, SW0IVD-CNVQ
- Macro library package — SW0NX-MSDQ, SW0IVD-MSDQ, SW0NX-MSPQ, SW0IVD-MSPQ
- Ladder sequence generation and connection package — SW0NX-LNKQ, SW0IVD-LNKQ

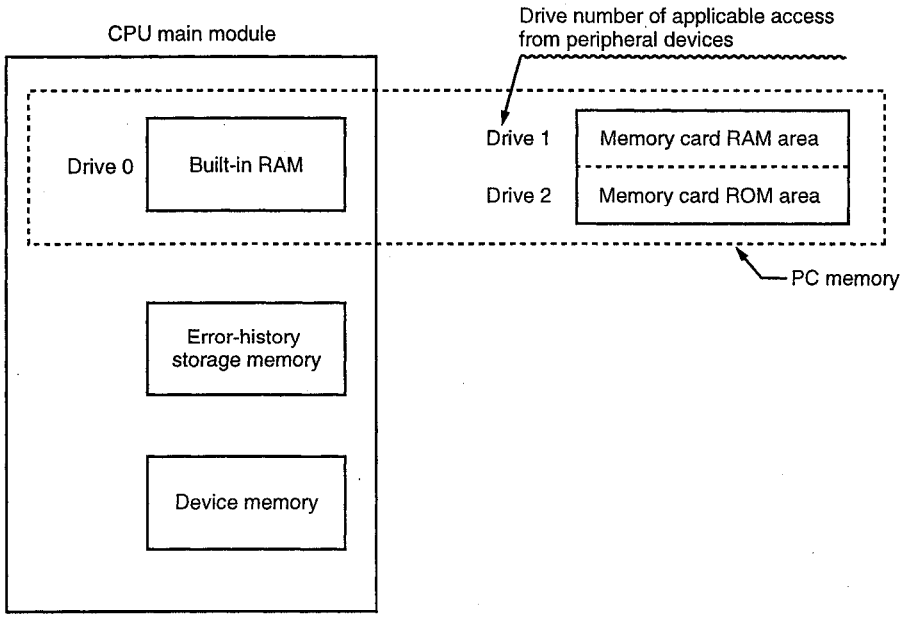
Remarks

The following peripheral devices and software packages cannot be used by Q2ASCPU.

- A□PU (Programming module)
- A6WU (ROM writer module)
- A6DU-B (Data access module)
- A6TEL (Model interface module)
- A6GPP (Intelligent GPP)
- A6HGP (Handy graphic programmer)
- A6PHP (Plasma handy graphic programmer)
- A7PHP (Plasma handy graphic programmer)
- A7HGP (Handy graphic programmer)
- A7LMS-[□□□], A75LMS-[□□□], A76LMS-[□□□] (FA controller)
- Software packages for ACPU system startup
  SW□□□-GPPA, SW□□□-SAP2
- Utility software packages for ACPU
  SW□□□-GPPATEL, SW□□□-CADIF, SW□□□-DRWA, SW□□□-FUNP, SW□□□-TSAP2

3.3.3 Q2ASCPU Memory Block Diagram

The following is the configuration of Q2ASCPU memory blocks.



- Built-in RAM:
  Memory for storing parameters and sequence programs.
- Error-history storage memory:
  Memory for storing error data.
- Device memory:
  Memory for storing device data.
- Memory card (RAM, ROM area):
  Memory for storing files and comments for parameters, sequence program, sampling trace, etc.
- PC memory:
  Refers to all memory of drives 0 to 2.

For the file types stored in each memory, refer to "File Types & Storage Destinations of Files Managed BY QnACPU" in the QnACPU Programming Manual (Fundamentals).

4 Performance Specification

Performance specification of Q2ASCPU module is as follows:

Item			Model				Remark
			Q2ASCPU	Q2ASCPU-S1	Q2ASHCPU	Q2ASHCPU-S1	
Control method			Repetitive operation of stored program				
I/O control method			Refresh mode				Direct I/O is enabled by specifying direct I/O (DX□, DY□).
Programming language			Sequence control dedicated language				
			Relay symbol language, logic symbolic language, MELSAP3 (SFC)				
Processing speed (sequence instructions) (u s / step)		LD	0.2		0.075		
		MOV	0.6		0.225		
Number of instructions (type)	Sequence instructions		39				
	Basic instructions		230				
	Application instructions		321				
	Special dedicated instructions		171				
Constant scan (ms) (function to make scan time constant)			5 to 2000 (configurable in multiple of 5 ms module)				Set parameter values to specify
Memory capacity			Capacity of loading memory cards (2036 kbyte maximum)				
Program capacity	Number of steps (steps)		28 k maximum	60 k maximum	28 k maximum	60 k maximum	
	Number of files (files)		28	60	28	60	
I/O device points (points)			8192 (X/Y0 to 1FFF)				Number of usable points in program
I/O points (points)			512 (X/Y0 to 1FF)	1024 (X/Y0 to 3FF)	512 (X/Y0 to 1FF)	1024 (X/Y0 to 3FF)	Number of points accessible to actual I/O modules
Number of device points	Internal relay [M] (points)		Default 8192 (M0 to 8191)				Set parameter values to specify usable points
	Latch relay [L] (points)		Default 8192 (L0 to 8191)				
	Link relay [B] (points)		Default 8192 (B0 to 1FFF)				
	Timer [T] (points)		Default 2048 (T0 to 2047) (for low-speed timer/high-speed timer speed) Select between low-speed timer/high-speed timer by instruction Set the measurement module of low-speed timer/high-speed timer by parameter (Low speed timer: 10 to 1000 ms, 10 ms/module, default 100 ms) (High speed timer: 1 to 100 ms, 1 ms/module, default 10 ms)				
	Retentive timer [ST] (points)		Default 0 (ST0 to 2047) (for both the low-speed and high-speed timers) Use instructions to switch from the low-speed timer to high-speed timer, and vice versa. Set the measurement unit for the high/low-speed timer using parameters. (Low-speed timer : 10 to 1000 ms, 10 ms increments, default 100 ms) (High-speed timer : 1 to 100 ms, 1 ms increments, default 10 ms)				
	Counter [C] (points)		• Normal counter default 1024 (C0 to 1023) • Interrupt counter maximum 48 (default 0 point, set via parameter)				
	Data register [D] (points)		Default 12288 (D0 to 12287)				
	Link register [W] (points)		Default 8192 (W0 to 1FFF)				
	Annunciator [F] (points)		Default 2048 (F0 to 2047)				
	Edge relay [V] (points)		Default 2048 (V0 to 2047)				
	File register [R] (points)		32768 (R0 to 32767), may use up to maximum of 1042432 points by block conversion				
			1042432 (ZR0 to 1042431) No block conversion necessary.				
	Special link relay [SB] (points)		Default 2048 SB0 to 7FF				Number of device points is fixed.
	Special link register [SW] (points)		Default 2048 SW0 to 7FF				
	Step relay [S] (points)		8192 (S0 to 8191)				
	Index register [Z] (points)		16 (Z0 to 15)				
	Pointer [P] (points)		4096 (P0 to 4095), set parameter values to select usable range of in-file pointer/shared pointers				
	Interrupt pointer [I] (points)		48 (10 to 47), set parameter values to select periodic interval of system interrupt pointers 128 to 131 (5 to 1000 ms, in 5 ms modules)				

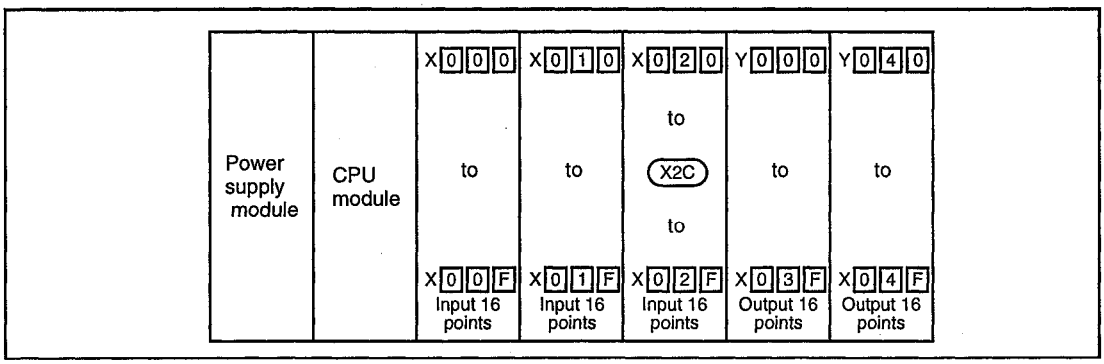
Item		Model				Remark
		Q2ASCPU	Q2ASCPU-S1	Q2ASHCPU	Q2ASHCPU-S1	
Number of device points	Special relay [SM] (points)	2048 (SM0 to 2047)				Number of device points is fixed
	Special register [SD] (points)	2048 (SD0 to 2047)				
	Function output [FX] (points)	16 (FX0 to F)				
	Function output [FY] (points)	16 (FY0 to F)				
	Function output [FD] (points)	5 (FD0 to 4)				
Link direct device		Device for direct access of link devices. For MELSECNET/10 use only. Specified form: J□□\□□				
Special function module direct device		Device for direct access of special function module buffer memory. Specified form: U□□\G□□				
Latch (power failure compensation) range		L0 to 8191 (default) (Latch range may be specified for B, F, V, T, ST, C, D, W)				Set parameter values to specify
Remote RUN/PAUSE contact points		May specify one RUN / PAUSE contact point each by X0 to 1FFF.				
Clock function		Year, month, date, hour, minute, second, day of week (auto-detects leap years) Accuracy: −1.7 to +4.9 s (TYP. +1.7 s) / d at 0 degrees centigrade Accuracy: −1.0 to +5.2 s (TYP. +2.2 s) / d at 25 degrees centigrade Accuracy: −7.3 to +2.5 s (TYP. −1.9 s) / d at 55 degrees centigrade				
Allowable momentary power failure period		By power supply module				According to Sec. 16.1.
5 VDC internal current consumption (A)		0.3	0.3	0.7	0.7	
Mass {kg (lb)}		0.5 (1.11)	0.5 (1.11)	0.5 (1.11)	0.5 (1.11)	
External dimension {mm (inch)}		130 X 54.5 X 110 (5.12 X 2.15 X 4.33)				

5 Allocation of I/O Numbers

This section describes how I/O numbers are allocated to facilitate data exchange with Q2ASCPU using I/O modules as well as special function modules.

5.1 What are I/O Numbers?

I/O numbers are used in sequence programs for receiving data from input modules and outputting data to output modules.
I/O numbers are expressed as three-digit hexadecimals.
The following is an example of I/O numbers when all I/O modules are 16 points modules.



I/O Number Scheme

Remarks

It is possible to enter I/O numbers as two-digit numbers when programming with peripheral devices for GPP function.

I/O number		Input by peripheral devices
X010	→	X10
Y020	→	Y20

5.2 I/O Number Allocation Scheme

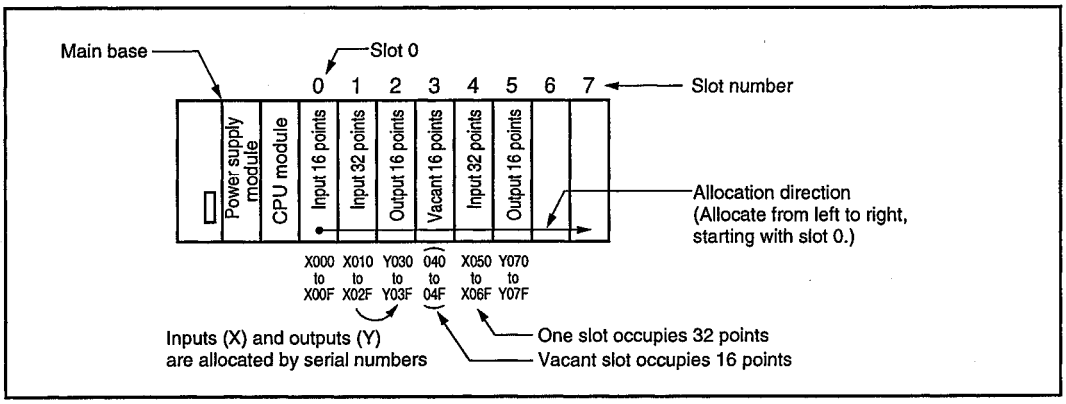
The following I/O numbers are allocated when the PLC is powered on or the CPU module is reset.

In the sequence program, specify I/O numbers that are allocated according to the following items.

(1) I/O numbers are allocated starting with 0 for slot 0 of the main base module (right side of the CPU module) going from left to right.

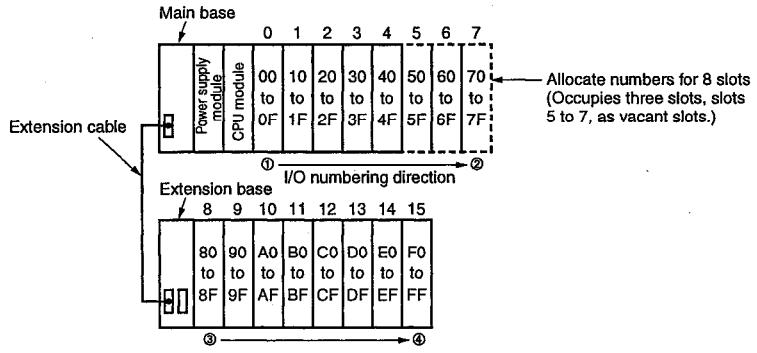
(2) The I/O module and special function modules attached to the base module occupy as many I/O numbers as there are I/O points for each module.

(3) Sixteen points are allocated for each vacant slot without an I/O module or special function module.



(4) If an extension base module is loaded, numbers are allocated starting with the next number following the main base modules number.

(5) The I/O numbers for the base module are allocated under the notion that there are eight slots altogether.
If a base module used is a five-slot type, numbers equivalent to three slots (48 points) are added to the last I/O number of the five-slot base module. The subsequent numbers are then allocated to the next extension base.



* I/O numbers are allocated in the order of ①→ ②→ ③ → ④.
Example above shows a case where all slots are 16 points modules.

5.3 I/O Allocation by GPP Function

Q2ASCPU can control I/O modules and special function modules without I/O allocation by GPP function.

I/O allocation by GPP function may be used most effectively under the following circumstances.

(1) Purpose of I/O allocation by GPP function

(a) When using a five-slot base module I/O points effectively used by allocating three slots to zero points.

(b) Reserving points for future system expansion of converting to module other than 16-point modules.

(c) Preventing I/O numbers from changing when removing a malfunction I/O module or special function module that is not a 16-point type.

(d) There is less frequent need for program-I/O number modification, since the I/O numbers of each base module can be changed one slot at a time to match the designed programs I/O numbers.

(2) Scheme of I/O allocation by GPP function

There are two methods used to allocate I/O numbers by GPP function.

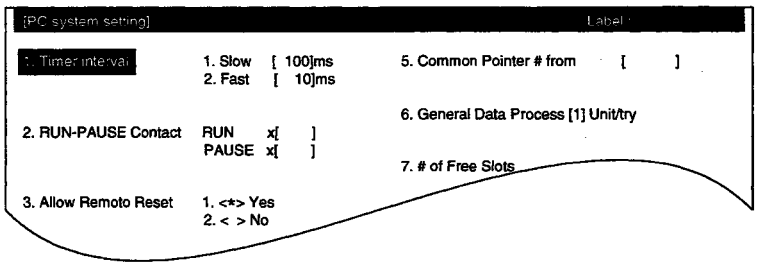
1) Allocate points to vacant slots in the main base and extension base (vacant slot points).

2) Specify I/O number allocation by slots for each main base and extension base.

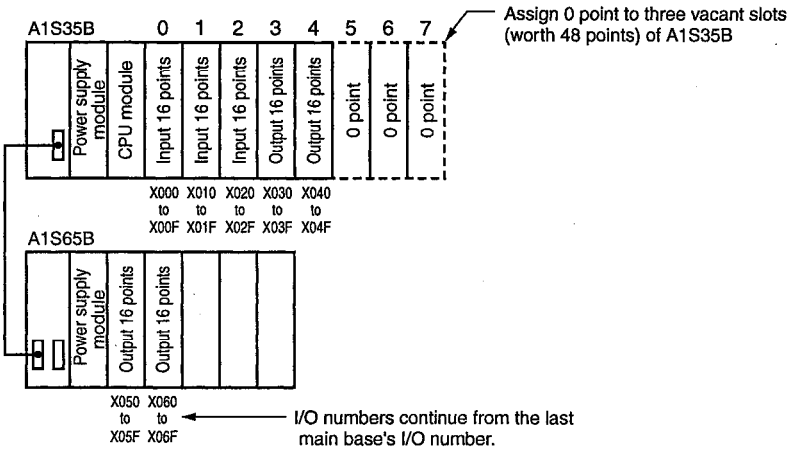
In either case, set parameter values to execute each allocation scheme. If both 1) and 2) are specified at the same time, the latter takes precedence.

(a) Setting the vacant slot points

The system specifies points to all vacant slots on the base. If a system has no parameter set, vacant slots are assigned 16 points each. Points are assigned on the PC System Setting screen under the parameter mode in 7. # of Free Slots.

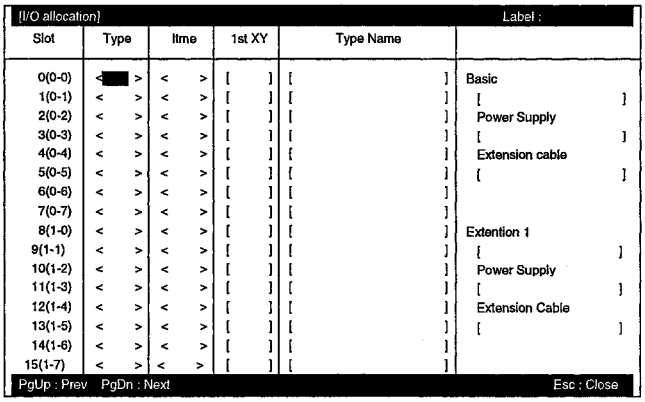


Points are set in 0 to 64 points range in modules of 16 points.
The default value is 16 points.
Example: When a vacant slot is assigned zero points.



(b) Setting of I/O number allocation

Specify the I/O number allocation by slots for each of main base and extension base.
Specify the I/O number allocation on the I/O allocation screen under the parameter mode.



The setting is done as below:

Item		Setting contents	Setting range	Default value
Slot specification		Set data for each slot. (No need to set all)	Vacant/input/output/special	Not specified
	Type	Set module type		
	Point	Set module points	0 to 64 points (in modules of 16 points)	
	Head XY	Set head XY number of module	0 to 1FFF (in modules of 16 points)	
	Model	Set model name of module	16 characters or less	
Base specification		Set data for each base	16 characters or less	Not specified
	Power supply module	Set model name of power supply module		
	Extension cable	Set model name of extension cable	16 characters or less	

If no setting is specified, the following rule applies

- Type, Point: Follows modules actually loaded
- Head XY : Numbers used are obtained by adding the necessary points to modules already set. An error (SP.MODULE LAY ERROR) is detected in case of duplicate numbers.

POINT

This poses no problem even if power supply module name was not specified under the base specification. The name is used only by the current-capacity check under the PC diagnostics mode, and is not used by the CPU module.

When the I/O allocation is set, the CPU module performs the following process:

1) For each slot of each base, one of the following allocations can be used:

Point allocation			
Vacant slot	Input module	Output module	Special function module
0	—	—	—
16	16	16	16
32	32	32	32
48	48	48	48
64	64	64	64

2) The I/O number allocated by GPP function takes precedence regardless of actual loading status of the modules.
   - If the numbers specified are less than the actual I/O module points, the actual usable points are reduced. For example, if the actual input module is a 32-point unit, and if the GPP function I/O allocation specifies 16 points, the second 16 points are not available for use.
   - If the numbers specified are more than the actual I/O module points only, points exceeding the actual module points become dummy points.
   - If an actual slot is specified as a vacant slot, such actual I/O module becomes unavailable for use.
3) Those slot to which points are not allocated by GPP function have points allocated according to the module points actually loaded.
4) The I/O slots not allocated points by GPP function are assigned numbers following the I/O numbers already allocated.

(3) Precaution

(a) If the parameter I/O allocation is different from the actual I/O module, normal I/O operation cannot be carried out.

Actual module	I/O allocation	Result
Input	Output	No input
Output	Input	No output
I/O	Special	CPU module error
Special	I/O	CPU module error

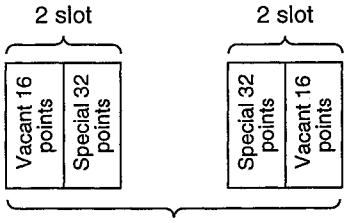
(b) The I/O allocation where a special function module is loaded must be equal to the actual setting.
If the setting is not the same, an error is generated.

1) A11VC ..... Special : 16 points

2) AI61 ..... Special : 32 points

3) AG62 ..... Input : Specified points

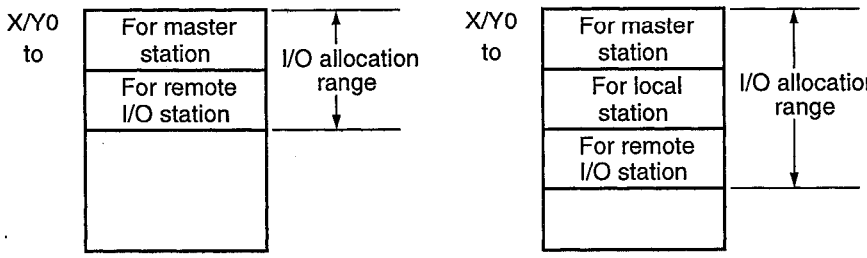
4) 2 slot occupied modules ... Set vacant 16 points and special 32 points



Refer to users' manual for special module being used.

(c) When the performing MELSECNET data link, allocate I/O as follows:

1) For the master station, it is necessary to allocate I/O for the master station and all remote stations.



2) For a local station, only the local stations I/O allocation needs to be performed.

3) An I/O mixed module (A42XY, etc.) should be allocated with an output module.

(d) When the MELSECNET/10 network is used, perform I/O assignment only for the host (master station.)
I/O assignment setting to a remote I/O station in MELSECNET/10 cannot be performed since it is unrelated.
When assigning I/Os to a remote I/O station in MELSECNET/10, do so using the network parameters.

REMARKS

For a remote I/O station of MELSECNET(II) and /B, the "I/O assignment setting" network parameters is unrelated. Therefore, the I/O assignment cannot be performed.

5.4 I/O Number Allocation Examples

This section gives examples of I/O numbers allocated when I/O allocation is made using the GPP function.

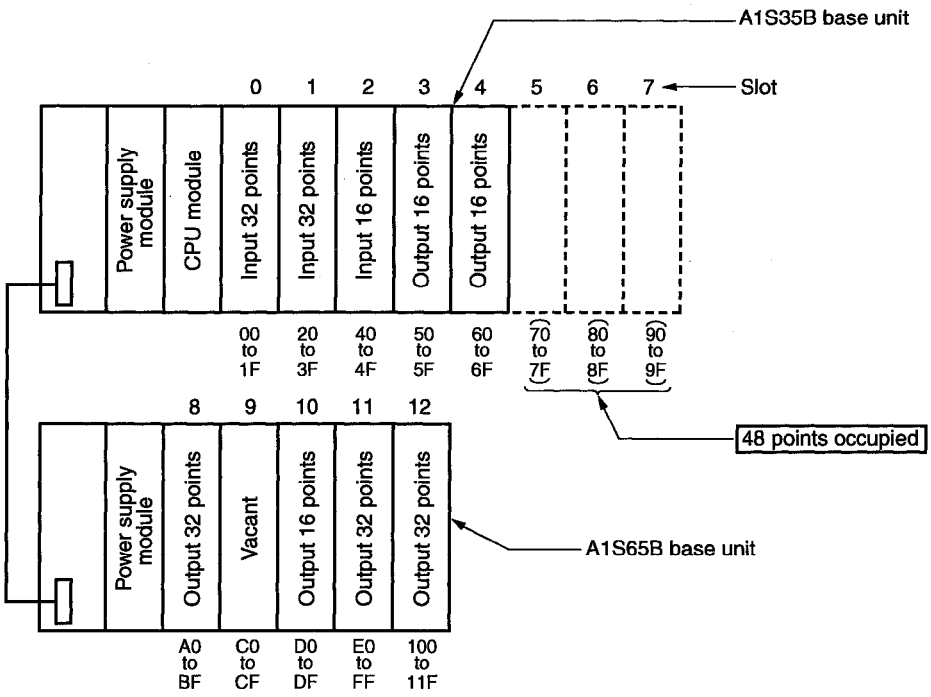
(1) When it is desired to change 16-point vacant slots into slots of 0 and 32 points

Since three vacant slots are made when the A1S35B is used, it is desired to change these slots into 0 points to increase the number of I/O points usable by the CPU module.

In addition, since a 32-point input module will be mounted to the currently vacant slot later, it is desired to reserve 32 I/O points to make the I/O number allocation unchanged.

To achieve the above, make the following I/O allocation.

(a) Mounting status and I/O numbers



(b) I/O numbers allocated when I/O allocation is made using GPP function

1) I/O allocation example

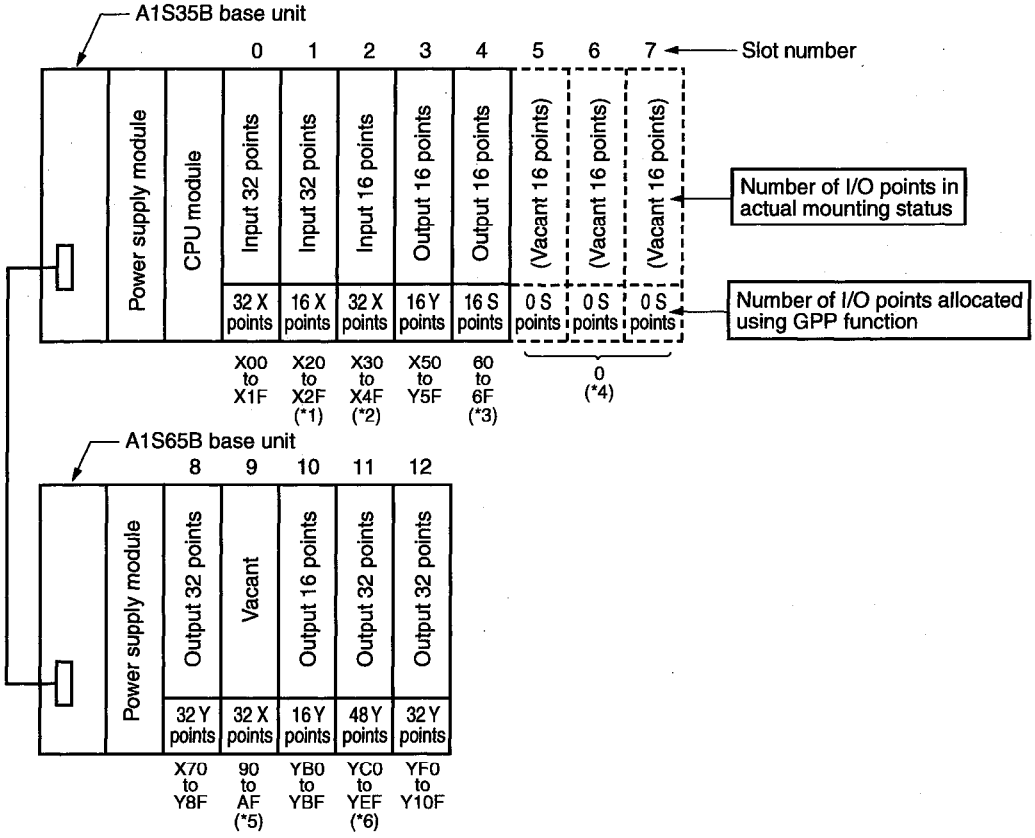
[I/O allocation] Label

Slot	Type	Itme	1st XY	Type Name	
0(0-0)	<Inp >	<32Pt>	[0]	[A1SX41]	Basic
1(0-1)	<Inp >	<16Pt>	[20]	[A1SX40]	[A1S35B]
2(0-2)	<Inp >	<32Pt>	[30]	[A1SX41]	Power Supply
3(0-3)	<Out >	<16Pt>	[50]	[A1SX40]	[A1S61P]
4(0-4)	<Free>	<16Pt>	[60]	[ ]	Extension Cable
5(0-5)	<Free>	< 0Pt>	[ ]	[ ]	[A1SC12B]
6(0-6)	<Free>	< 0Pt>	[ ]	[ ]	
7(0-7)	<Free>	< 0Pt>	[ ]	[ ]	
8(1-0)	<Out >	<32Pt>	[70]	[A1SY41]	Extention 1
9(1-1)	<Inp >	<32Pt>	[90]	[A1SX41]	[A1S65B]
10(1-2)	<Out >	<16Pt>	[B0]	[A1SY40]	Power Supply
11(1-3)	<Out >	<48Pt>	[C0]	[ ]	[A1S61P]
12(1-4)	<Out >	<32Pt>	[F0]	[A1SY41]	Extension Cable
13(1-5)	< >	< >	[ ]	[ ]	[ ]
14(1-6)	< >	< >	[ ]	[ ]	
15(1-7)	< >	< >	[ ]	[ ]	

PgUp: Prev PgDn: Next Esc: Close

I/O allocation example using GPP function

2) I/O numbers allocated after I/O allocation made using GPP function



Remark

*1: Since the setting is 16 points, the latter-half 16 input points cannot be used.

*2: Since the setting is 32 points, 40 to 4F are dummy points.

*3: Since the setting is 16 vacant (S) points, the slot cannot be used for output.

*4: Since the setting is 0 vacant (S) points, the three slots are not allocated I/O points.

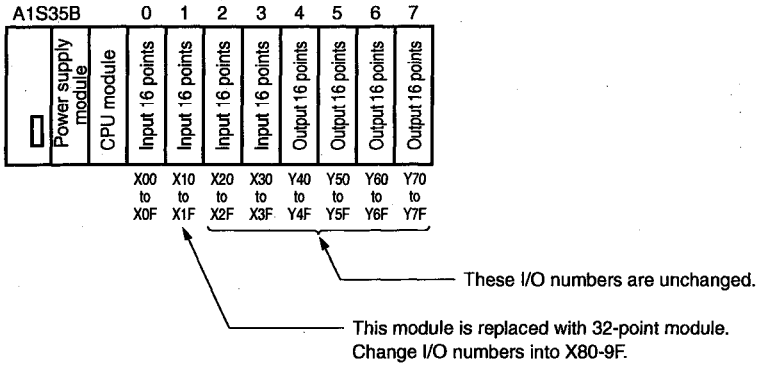
*5: Since the setting is 32 input (X) points, the slot is used for 32-point input.

*6: Since the setting is 48 points, E0 to EF are dummy points.

(2) When it is desired to replace 16-point input module with 32-point input module

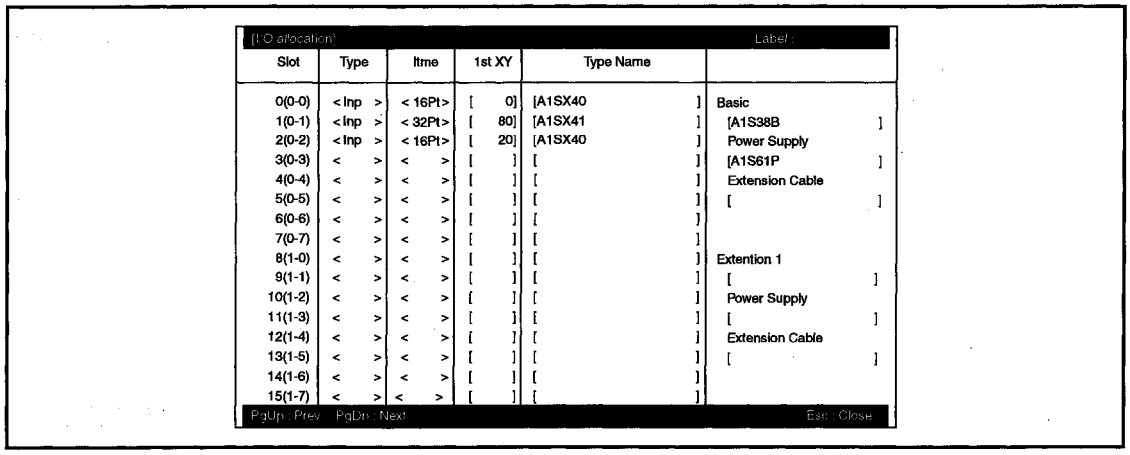
It is desired to replace a 16-point input module with a 32-point input module without changing the whole I/O number allocation in a system designed using 16-point input modules.

(a) Before-replacement mounting status and I/O numbers



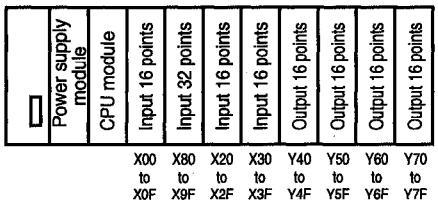
(b) I/O numbers allocated when I/O allocation is made using GPP function

1) I/O allocation example


[I/O allocation]
Label
Slot
Type
Itme
1st XY
Type Name
0(0-0) < Inp > < 16Pt > [0] [A1SX40] Basic
1(0-1) < Inp > < 32Pt > [80] [A1SX41] [A1S38B]
2(0-2) < Inp > < 16Pt > [20] [A1SX40] Power Supply
3(0-3) < > < > [ ] [ ] [A1S61P]
4(0-4) < > < > [ ] [ ] Extension Cable
5(0-5) < > < > [ ] [ ] [ ]
6(0-6) < > < > [ ] [ ]
7(0-7) < > < > [ ] [ ]
8(1-0) < > < > [ ] [ ] Extention 1
9(1-1) < > < > [ ] [ ] [ ]
10(1-2) < > < > [ ] [ ] Power Supply
11(1-3) < > < > [ ] [ ] [ ]
12(1-4) < > < > [ ] [ ] Extension Cable
13(1-5) < > < > [ ] [ ] [ ]
14(1-6) < > < > [ ] [ ]
15(1-7) < > < > [ ] [ ]
PgUp : Prev PgDn : Next
Esc : Close

I/O allocation example using GPP function

2) I/O numbers allocated after I/O allocation is made using GPP function and module replacement is performed



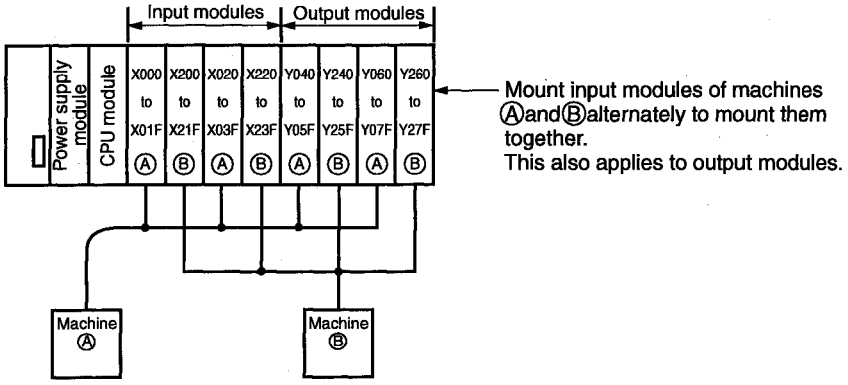
POINT

When "1st XY" has been set on the I/O allocation screen to change the I/O numbers, also set "1st XY" of the module next to the new module to keep the I/O numbers of the next module and later unchanged.
In the above example, since "1st XY" of Slot 2 was set for "20", consecutive I/O numbers starting with X30 are allocated to Slot 3 and later.

(3) When it is desired to mount separately-numbered input and output modules together on base unit

When one PLC is used to control machine Ⓐ (I/O No. X0 to X3F, Y40 to 7F) and machine Ⓑ (I/O No. X200 to X23F, Y240 to 27F), it is desired to mount input and output modules together on a base unit. In this case, make I/O allocation as shown below.

(a) Mounting status and desired I/O numbers



(b) Example of I/O allocation made using GPP function

[I/O allocation] Label:

Slot	Type	Itme	1st XY	Type Name	
0(0-0)	<Inp >	< 32Pt>	[0]	[A1SX41]	Basic
1(0-1)	<Inp >	< 32Pt>	[200]	[A1SX41]	[A1S38B]
2(0-2)	<Inp >	< 32Pt>	[20]	[A1SX41]	Power Supply
3(0-3)	<Inp >	< 32Pt>	[220]	[A1SX41]	[A1S61P]
4(0-4)	<Out >	< 32Pt>	[40]	[A1SY41]	Extension Cable
5(0-5)	<Out >	< 32Pt>	[240]	[A1SY41]	[ ]
6(0-6)	<Out >	< 32Pt>	[60]	[A1SY41]	
7(0-7)	<Out >	< 32Pt>	[260]	[A1SY41]	
8(1-0)	< >	< >	[ ]	[ ]	Extention 1
9(1-1)	< >	< >	[ ]	[ ]	[ ]
10(1-2)	< >	< >	[ ]	[ ]	Power Supply
11(1-3)	< >	< >	[ ]	[ ]	[ ]
12(1-4)	< >	< >	[ ]	[ ]	Extension Cable
13(1-5)	< >	< >	[ ]	[ ]	[ ]
14(1-6)	< >	< >	[ ]	[ ]	
15(1-7)	< >	< >	[ ]	[ ]	

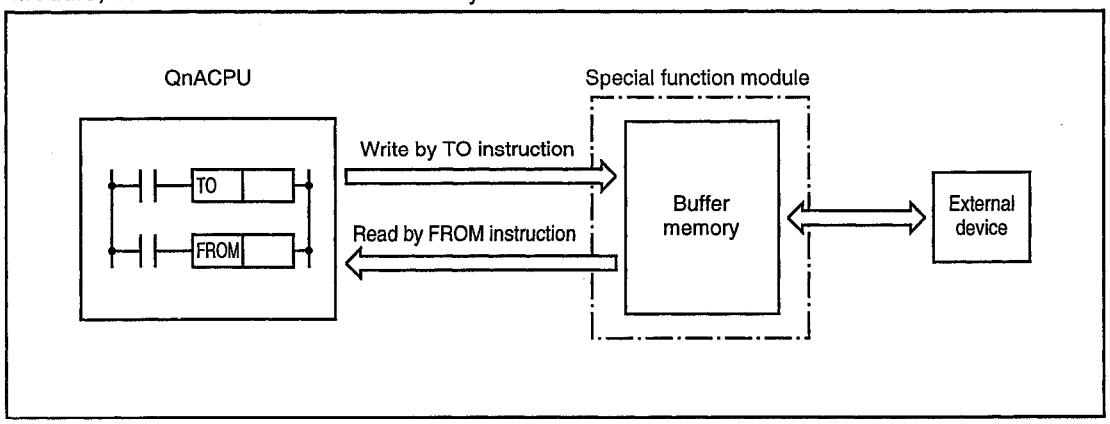
PgUp : Prev PgDn : Next Esc : Close

6 Data Exchange with Special Function Modules

This section describes reading data from, and writing data to, special function modules of Q2ASCPU.
The special function module enables Q2ASCPU to handle analog values and high-speed pulse, which cannot be handled by the I/O modules alone.
For example, analog values are converted to digital values by the analog/digital conversion module for Q2ASCPU use.
The special function modules are equipped with memories (buffer memories) to store input data obtained from external sources and data for external output.
There are two ways to perform data read/write from the special function module of Q2ASCPU:
(1) Use the FROM/TO instructions
(2) Use special direct device
Each of these items is described below.

6.1 Reading and Writing Data from Q2ASCPU by FROM/TO Instruction

Executing the FROM/TO instructions reads data from buffer memory of the special function module, or writes data to buffer memory.



Data exchange with special function module

The FROM instruction stores data read from the buffer memory into the specified device. The TO instruction writes data of the specified device to the buffer memory.

Remarks

(1) For details of FROM/TO instruction, refer to QCPU(Q mode)/QnACPU PROGRAMMING MANUAL (Common Instructions).

(2) For details of the buffer memory of the special function module, refer to the manual for the particular special function module being used.

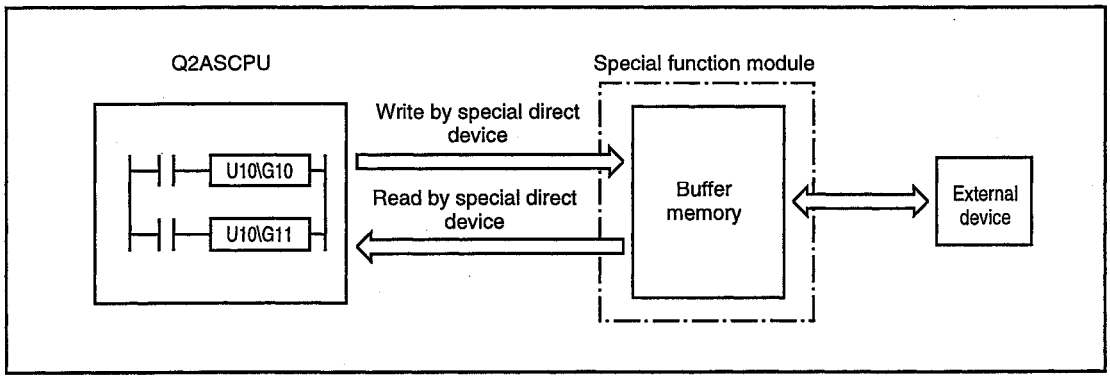
POINT

If a FROM/TO instruction is executed for a special function module frequently in a short scan time, the objective special function module may fail to process correctly.

To execute a FROM/TO instruction for a special function module, set the execution intervals meeting the processing and conversion time of that module using the timer and the constant scan function of it.

6.2 Reading and Writing Data from Q2ASCPU by Special Direct Device

The special direct device, like the FROM/TO instructions, reads data stored in the buffer memory of the special function module, or writes data to the buffer memory.



The special direct device represents the buffer memory of the special function module as the Q2ASCPU device.

(Example) U10\G10:

U10 → Head I/O No. 100 of the special function module (hexadecimal).

G10 → Address 10 of buffer memory of the special function module (decimal).

Remarks

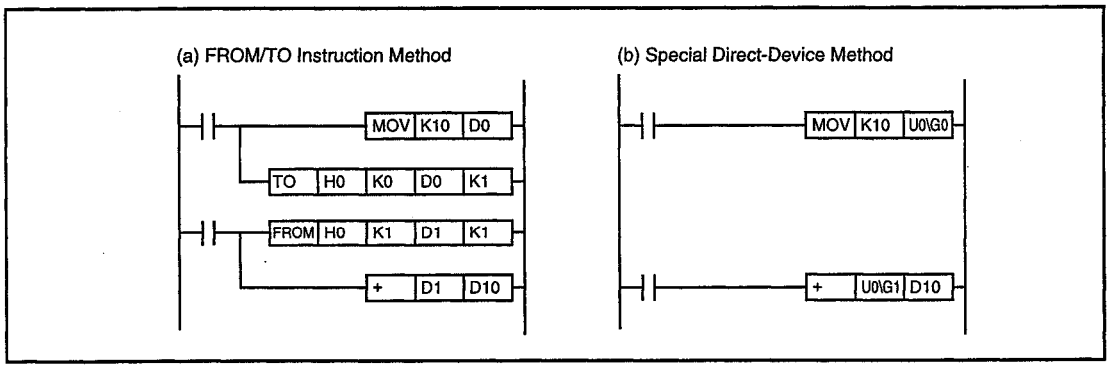
For details of the special direct device, refer to QnACPU Programming Manual (Fundamentals).

Unlike the FROM/TO instructions, the CPU module can treat the buffer memory of a special function module as a direct device.

With this the total number of program steps can be reduced. The processing time, however, is the same as the FROM/TO instructions.

[Example]

Write data into address 0 and read data from address 1 of the buffer memory of the special function module loaded to X\Y0.



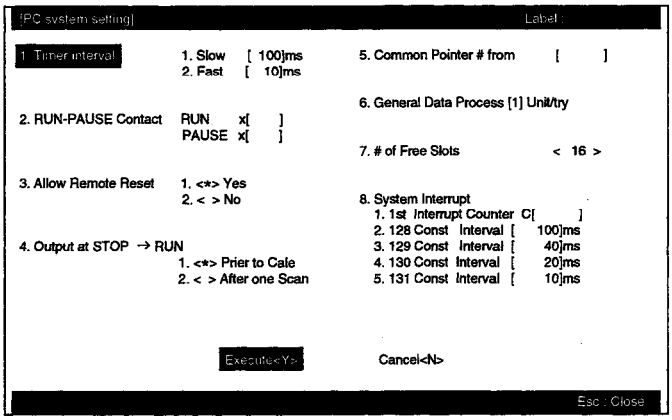
POINT

If the special function module data must be read frequently, rather than using the special direct device for each instruction it is better to use the FROM instruction once to read the data, store it in the data register, and access it later.

The programs scan time will be longer when using the special direct device, since the special function module must be accessed for each instruction.

6.3 Processing a Data-Communication Request from the Special Function Module

When a data-communication request is received from a special function module, such as serial-communication module, Q2ASCPU processes the request during the END process.
By specifying parameters, Q2ASCPU can process all data-communication requests received during one scan through one END process. In this case, there is no data delay to each module. The END process, however, will take longer process data communication requests.
Batch processing of data-communication requests is set in Section 6. "General Data Process [1] Module/Try" on the "PC System Setting" screen under the GPP function parameter mode. 1 to 6 modules can be specified as the number of modules.


[PC system setting]
Label :
1. Timer interval
1. Slow [100]ms
2. Fast [10]ms
2. RUN-PAUSE Contact
RUN x[ ]
PAUSE x[ ]
3. Allow Remote Reset
1. <*> Yes
2. < > No
4. Output at STOP → RUN
1. <*> Prier to Cale
2. < > After one Scan
5. Common Pointer # from [ ]
6. General Data Process [1] Unit/try
7. # of Free Slots < 16 >
8. System Interrupt
1. 1st Interrupt Counter C[ ]
2. 128 Const Interval [100]ms
3. 129 Const Interval [40]ms
4. 130 Const Interval [20]ms
5. 131 Const Interval [10]ms
Execute<Y>
Cancel<N>
Esc : Close

7 Setting Automatic Refresh

7.1 For MELSECNET/MINI-S3

In order to automatically exchange batch refresh communication data of A1SJ71PT32-S3 master module (master module, hereinafter) with buffer memory area, set parameters like MELSECNET/MINI-S3 link information and I/O storage device.
Set these parameters in MELSECNET/MINI specification under GPP function parameter mode. By setting MELSECNET/MINI, I/O devices allocated to communication can be used directly in the sequence program. (FROM/TO instructions are not necessary).

POINTS

(1) Maximum of eight master modules can be set for automatic refresh in parameter setting. To communicate with nine or more modules, process 9th module and beyond through the FROM/TO instruction.

(2) Communication data for partition refresh I/O modules and communication data for remote terminal modules No. 1 to No. 14 cannot be processed through automatic refresh. Use the FROM/TO instructions for these cases.
- AJ35PTF-R2 RS-232C interface module
- AJ35PT-OPB-M1-S3 mount type operation box
- AJ35PT-OPB-P1-S3 portable type operation box

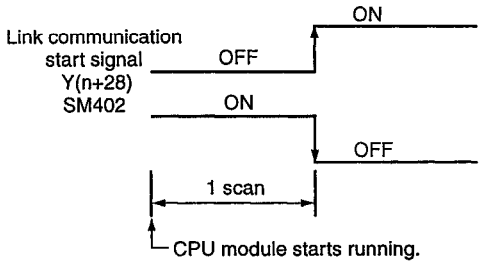
(3) Q2ASCPU automatically turns on the link communication start signal Y(n+18) / Y(n+28) for master modules specified for automatic refresh. Therefore there is no need to turn these on via sequence program.

(4) Automatic-refresh process of I/O data is batch processed after executing the Q2ASCPU END instruction. (Automatic-refresh process is performed when the CPU module is in RUN/PAUSE/STEP RUN status.)

(5) Depending on the remote terminal module to be connected, master module may perform this process when link communication start signal Y(n+18) / Y(n+28) is turned off.

For example, when using AJ35PTF-R2 RS-232C interface module under no-protocol mode, parameters must be written into parameter area (buffer memory address 860 to 929) while link communication start signal is turned off.

Parameters need to be written during the first scan since the link communication start signal turns on after the first scan while in the CPU RUN state.



(6) Q2ASCPU does not perform automatic-refresh processing if hardware error message X(n+0)/X(n+20) or ROM error message X(n+8)/X(n+28) is generated from the master module, which the automatic-refresh processing was specified.

(7) Set the data so that receive-data-refresh device and send-data-refresh device do not overlap.

(1) The following table shows automatic refresh parameter specification items, setting range, description, etc. Buffer memory addresses of master modules communicating with Q2ASCPU are also shown.
Parameters are set for all master modules being used.

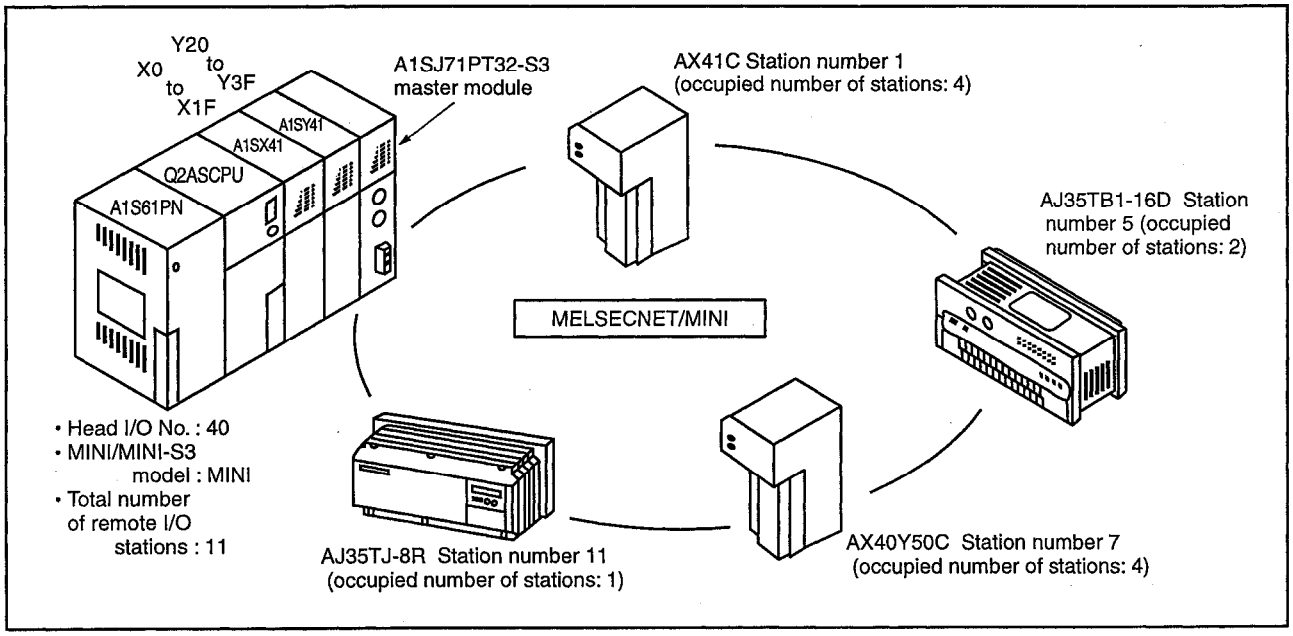
Master module I/O signal	Master module buffer memory address	Item	Setting range	Description	Default
—	—	Number of master modules	0, 1 to 8 units	- Sets a total number of master module being used. "0" indicates no automatic refresh.	Uses values specified in I/O allocation under parameter mode *3
—	—	Head I/O No.	Number of CPU module I/O points	- Sets head I/O number where master module is installed.	
—	—	MINI/MINI-S3 model	MINI or MINI-S3	- MINI . . . . . For dedicated I/O mode (occupies 32 points) - MINI-S3 . . . . For extension mode (occupies 48 points)	
—	0	Total number of remote I/O stations	0 to 64 stations	- Set only when MINI is specified - Setting is not necessary when MINI-S3 is specified, since the number of initial ROM master modules will be used. (Any setting here will be ignored in this case.)	
—	110 to 141	*4 Receive data storage device	- X - M , L, B, T, ST, C, D, W, R, ZR, none (bit devices in multiple of 16)	- Sets device to store receive/send data for batch refresh - Setting specifies head number of device - Starting from device head, occupies automatic refresh area for specified number of stations (8 points/station X 64 stations = 512 points ... bit device) *2 - Use of X/Y remote I/O range device is recommended	X1000 to X11FF
—	10 to 41	Send data storage device	- Y - M , L, B, T, ST, C, D, W, R, ZR, none (bit devices in multiple of 16)		Y1000 to Y11FF
—	1	Number of retries	0 to 32 times	- Sets number of retries in case of communication error - Error is not generated if communication recovers within number of specified retries	5 times
Y(n+1A) *1	—	FROM/TO response	Link priority or CPU priority (Priority selection of access to master module buffer memory.)	(1) Link priority . . Link access of MINI-S3 has priority. FROM/TO waits during link access - Possible to read received data which was refreshed at the same time - Creates FROM/TO wait time equal to maximum of {0.3 ms + 0.2 ms (number of division-refresh stations} (2) CPU priority . . Access of FROM/TO instruction from CPU module has priority. Performs interrupt access even during link access. - Depending on timing, may read receive data during I/O refresh - No wait time for FROM/TO instruction	CPU priority
Y(n+1B) *1	—	Communication-error station data clear	Retention or clear (receive data)	- Retention . . . . Holds receive data for batch refresh and partition refresh - Clear . . . . . Set all points to OFF state.	Clear
—	100 to 103 195	Error station detection	M, L, B, T, ST, C, D, W, R, ZR, none (bit devices in multiple of 16)	- Sets head device that stores error station detection data - MINI ... occupies 4 words, MINI-S3 ... occupies 5 words	None
—	107 196 to 209	Error No.	T, ST, C, D, W, R, ZR	- Sets head device that stores error codes when an error is generated - MINI ... occupies 1 word, MINI-S3 ... occupies (1 + number of remote terminals) words	None
—	4	Circuit error check	- Send test message (test) - Send OFF data (OFF) - Send data prior to circuit error (retention)	- Sets data communication method to verify error location in case of a line error.	Retention
—	—	CPU STOP action	Stop or continue	- Sets action status for CPU module STOP situation.	Stop

*1. "n" is determined by mounting position of master module.
*2. If total number of remote I/O stations is an odd number, each storage device occupies space for one additional station.
*3. If the number of master modules is set to blank in parameter setting, automatic refresh can be performed without the above specification. It is necessary, however, to register model name in I/O allocation.
*4. When specifying the inputs (X) as the receive data storage device, use the I/O numbers, which are used for the module mounted on the main or extension base, or later.
If the I/O number range used for the module mounted on the main or extension base is used for the I/O of the receive data storage device, the CPU module imports both the ON/OFF data of the inputs from the input module and the ON/OFF data from automatic refresh of MELSECNET/MINI-S3. Hence, the inputs (X) of the CPU module go wrong.

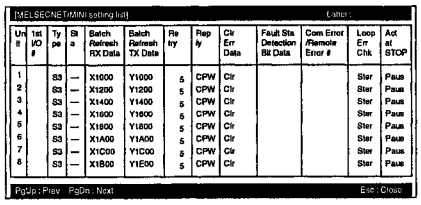
(2) Following system demonstrates how to set receive/send data storage devices.

[Example]

In case when devices after X/Y400 are used as remote I/O stations:

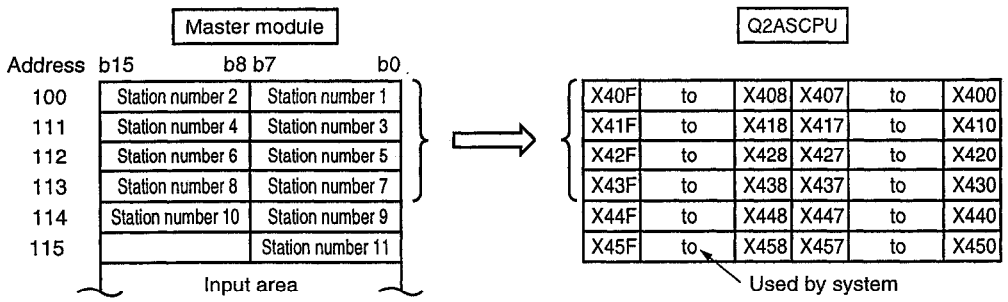


In the system example above, GPP function are set as follows:



In the system example above, receive/send data storage devices are as follows:

(a) Receive data storage device

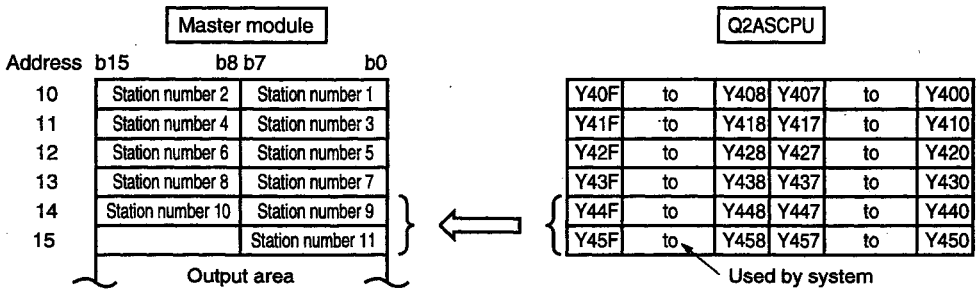

Master module
Q2ASCPU
Address b15 b8 b7 b0
100 Station number 2 Station number 1
111 Station number 4 Station number 3
112 Station number 6 Station number 5
113 Station number 8 Station number 7
114 Station number 10 Station number 9
115 Station number 11
Input area
X40F to X408 X407 to X400
X41F to X418 X417 to X410
X42F to X428 X427 to X420
X43F to X438 X437 to X430
X44F to X448 X447 to X440
X45F to X458 X457 to X450
Used by system

1) For receive data storage device, set the device number (X400) of b0 in station number 1.

2) Receive data storage device can occupy X400 to X45F.
One additional station is used in the system example above, since the total number of stations is an odd number.

3) Device numbers of connected input modules are as follows:
Station numbers 1 to 4 AX41C → X400 to X41F
Station numbers 5 to 6 AJ35TB-16D → X420 to X42F
Station numbers 7 to 8 AX40Y50C → X430 to X43F
X440 to X45F are also refreshed at the same time and will always be OFF
X440 to X45F should not be used for sequence programs.

(b) Send data storage device


Master module
Q2ASCPU
Address b15 b8 b7 b0
10 Station number 2 Station number 1
11 Station number 4 Station number 3
12 Station number 6 Station number 5
13 Station number 8 Station number 7
14 Station number 10 Station number 9
15 Station number 11
Output area
Y40F to Y408 Y407 to Y400
Y41F to Y418 Y417 to Y410
Y42F to Y428 Y427 to Y420
Y43F to Y438 Y437 to Y430
Y44F to Y448 Y447 to Y440
Y45F to Y458 Y457 to Y450
Used by system

1) For send data storage device, set the device number (Y400) of b0 in station number 1.

2) Receive data storage device can occupy Y400 to Y45F.
One additional station is used in the system example above, since the total number of stations is an odd number.

3) Device numbers of connected output modules are as follows:
Station numbers 9 to 10 AX40Y50C → Y440 to Y44F
Station number 11 AJ35TJ-8R → Y450 to X457

Y400 to Y43F and Y458 to Y45F are also refreshed at the same time
No output is performed, however.

POINTS

(1) If a same device is used as send data storage device and as receive data storage device, be sure that device numbers do not overlap.

If B0 is set as receive data storage device in the system configuration (example above) then B0 to B5F device range is occupied.

Set B60 or higher as send data storage devices.

If the send data storage device was set to B60, device range becomes B60 to BBF.

(2) When bit device is specified as send/receive data storage device, be sure to specify multiple of 16 as device numbers.

[Example] X0, X10, ... X100 .....
M0, M16,..... M256, ...
B0, B10, ... B100,.....

(3) Usable device range is (8 points) X (number of stations)

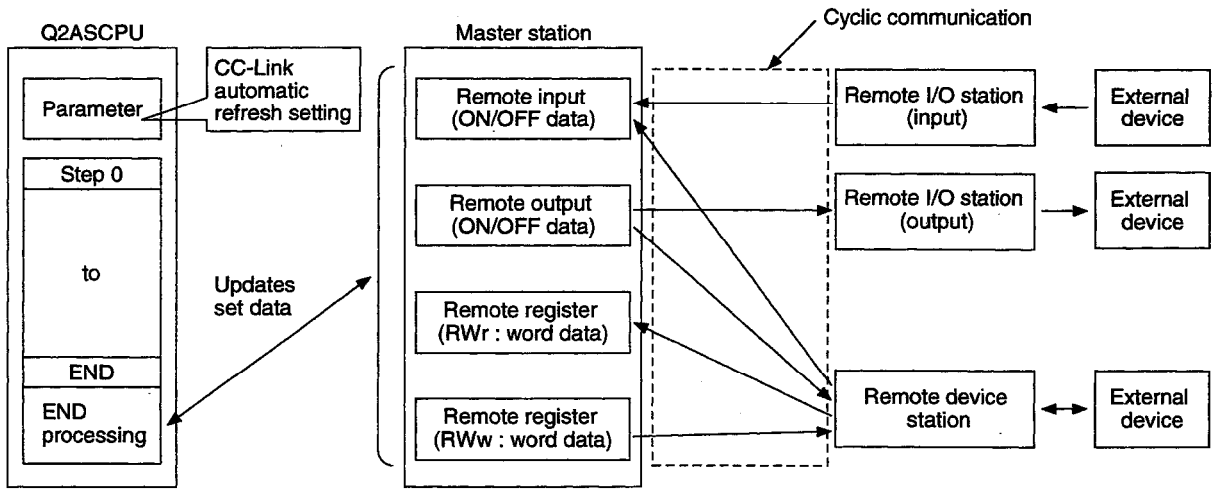
In case the number of stations is an odd number, 8 more points are necessary.

7.2 CC-Link Automatic Refresh Setting

The CC-Link automatic refresh is the automatic communication between Q2ASCPU and the buffer memory for cyclic communication of the master station or local station on the CC-Link. The data that can be communicated varies depending on the remote stations being connected.

- Remote I/O station (ON/OFF data communication)
- Remote device station (ON/OFF data + word data communication)
- Intelligent device station (ON/OFF data + word data communication)
- Master station/local station (ON/OFF data + word data communication)

Setting the CC-Link automatic refresh enables the use of FROM/TO instructions, which in turn allows for a direct communication with CC-Link's other stations without communicating with CC-Link master station.



(1) Settings to perform automatic refresh

Table 7.1 describes the data items to be set when executing automatic refresh function using Q2ASCPU.

Table 7.1 List of automatic refresh setting data

Item	Description	Setting range	Set station		
			M	L	T
Number of modules	Sets a number of CC-Link modules.	1 to 8	○	○	○
Module head I/O number	Sets a head I/O number for the CC-Link module.	000H to 0FE0H	○	○	○
Module type	Sets the type of CC-Link module installed (master station, local station, or standby station)	• M : master station • L : local station • T : standby station	○	○	○
Batch-refresh bit device for receiving data (Input data)	• Sets the device in which the batch refresh data received from a remote station is stored. • Upon setting the head device number, establishes a number of points corresponding to the number of specified stations (total number of stations), and refreshes the entire area. Also refreshes the output module area. • Use 16 points per unit for setting.	X, M, L, B, T, ST, C, D, W, R, ZR*	○	○	○
Batch-refresh bit device for transmitting data (Output data)	• Sets the device in which the batch refresh data transmitted to a remote station is stored. • Upon setting the head device number, establishes a number of points corresponding to the number of specified stations (total number of stations), and refreshes the entire area. Also refreshes the input module area. • Use 16 points per unit for setting.	Y, M, L, B, T, ST, C, D, W, R, ZR*	○	○	○
Batch-refresh word device for receiving data (Remote device: RWr)	• Sets the device in which the batch refresh data received from a remote station is stored. • Upon setting the head device number, establishes a number of points corresponding to the number of specified stations (total number of stations), and refreshes the entire area. Also refreshes the input/output module area. • Use one point per unit for setting.	M, L, B, T, ST, C, D, W, R, ZR*	○	○	○
Batch-refresh device for transmitting data (Remote device: RWw)	• Sets the device in which the batch refresh data transmitted to a remote station is stored. • Upon setting the head device number, establishes a number of points corresponding to the number of specified stations (total number of stations), and refreshes the entire area. Also refreshes the input/output module area. • Use one point per unit for setting.	M, L, B, T, ST, C, D, W, R, ZR*	○	○	○
Receiving buffer specification for transient station	• Sets capacities of the receiving buffer for each transient station.	80 to 4096	○	×	×
Transmission buffer specification for transient station	• Sets capacities of the transmitting buffer for each transient station.	80 to 4096	○	×	×
Special relay batch refresh device	• Sets the destination device of special relay.	M, L, B, T, ST, C, D, W, R, ZR*	○	○	○
Special register batch refresh device	• Sets the destination device of special register.	T, ST, C, D, W, R, ZR*	○	○	○

Remarks

1) For the set station in Table 7.1, M indicates the master station, while L and T denote the local and standby stations, respectively.
2) In Table 7.1, ○ indicates that setting is allowed, and × indicates that setting is not required.
3) * : R and ZR can be used as the automatic refresh device only when the file register (within its parameter) is set as "Using a specified file."
R and ZR cannot be used if "Using the same file as for the program" is set.

Table 7.1 List of automatic refresh setting data (Continued)

Item	Description	Setting range	Set station		
			M	L	T
Automatic update buffer specification	• Sets the capacity for an automatic update buffer.	128 to 4096	○	×	×
Number of total slave stations	• Sets the last station number of remote stations connected to a master station.	1 to 64	○	×	×
Delay timer	• Sets a delay time for link scan.	1 to 100 (0 is invalid)	○	×	×
Standby station specification	• Sets whether the standby master function is being used.	• Not used • Used	○	×	×
Number of retries	• Sets the number of retries when an transient transmission error occurs.	1 to 7	○	×	×
Number of automatic return stations	• Sets the number of automatic return stations in one link scanning cycle.	1 to 10	○	×	×
Operation specification when CPU is down	• Sets either "to continue" or "stop" the data link when the CPU module is shut off.	• Stop • Continue	○	×	×
Scan mode setting	• Sets synchronization/non-synchronization for the CPU module scan.	• Non-synchronization • Synchronization	○	×	×
Station type	• Sets device for each remote station.	• Remote I/O station • Remote device station • Intelligent device station	○	×	×
Number of occupied stations	• Sets the number of occupied stations for each remote station.	• 1 station • 2 stations • 3 stations • 4 stations	○	×	×
Reserved station specification	• Specifies whether or not the remote station is reserved.	• Not used • Used	○	×	×
Invalid station specification	• Sets valid/invalid status for detecting errors from remote stations.	• Invalid • Valid	○	×	×

Remarks

1) For the set station in Table 7.1, M indicate

2) In Table 7.1, ○ indicates that setting is allowed, and × indicates that setting is not required.

(2) Precaution

(a) The CC-Link automatic refresh function can be performed when both the Q2ASCPU and CC-Link module are function version "B."
If the function version of either the Q2ASCPU or CC-Link module is not "B," CC-Link automatic refresh function will not be performed.

(b) Up to 8 pieces of CC-Link modules can be set with the automatic refresh setting.
If more than 9 pieces of CC-Link modules are used, use the FROM/TO instruction within the sequence program, starting from the ninth module.

(c) When both the CC-Link module and MELSECNET/MINI-S3 master module are installed, but the automatic refresh operation has not been performed, set the default parameters in the MELSECNET/MINI-S3 master station module.

(d) By using the COM instruction and the G(P).ZCOM instruction, the automatic refresh operation can be performed to a CC-Link module while the sequence program is executed.
However, the automatic refresh operation cannot be performed to a CC-Link module using the J(P). ZCOM instruction. If used, an error code "4102" will be displayed (Network number specified by the network dedicated instruction does not exist).

(e) Refresh operation when MELSECNET(/10, /II) and MELSECNET/MINI-S3 coexist.
- Refresh operation will be performed in the order of MELSECNET(/10, /II) → CC-Link → MELSECNET/MINI-S3.
Due to this order of operation, if there are two (or more) sets of input data specifying the same area, the latter one (the data that was executed last) will overwrite the first one.
- Output data are output to MELSECNET(/10, /II), CC-Link, and MELSECNET/MINI-S3.

(f) The operations of Q2ASCPU when a CC-Link module is online/offline are described in the table below.

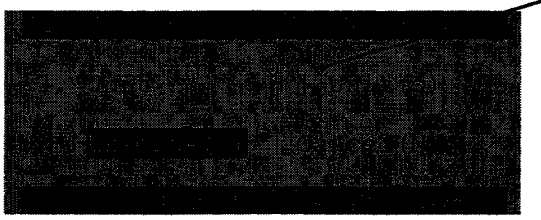
Automatic refresh parameter setting	CC-Link module status	Operations of Q2ASCPU
Set	Online	Communicates with a remote station using the specified automatic refresh parameter.
	Offline	The Q2ASCPU will generate no errors, but will not communicate with a remote station.
Not set	Online	Communicates with a remote station using FROM/TO instructions.
	Offline	The Q2ASCPU will generate no errors, but will not communicate with a remote station.

(g) Use the peripheral device listed below for the CC-Link automatic refresh setting.
- IBM PC/AT or 100% compatibles: SW2IVD-GPPQ type GPP function software package, GX Developer

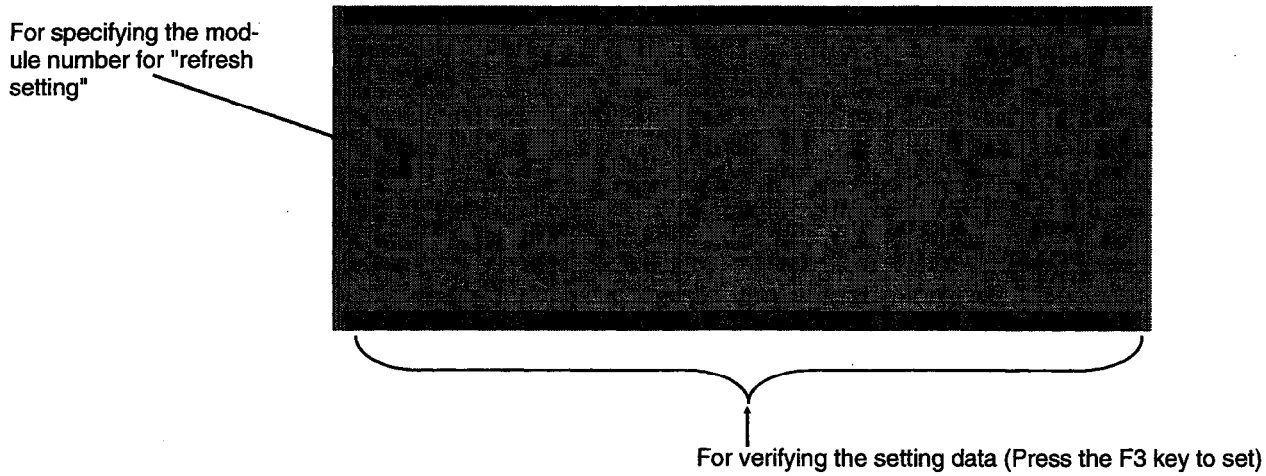
(3) Method to set

Use the following procedure for the CC-Link automatic refresh setting.

(a) Select CC-Link setting using the parameter to display the "CC-Link Setting window."

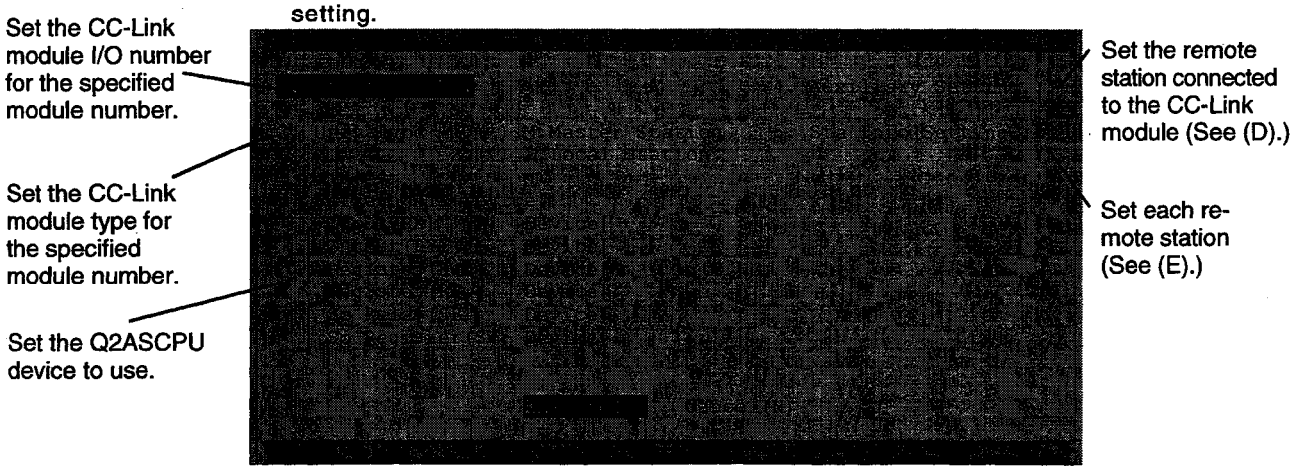


(b) To display the "CC-Link Setting List window," set the number of CC-Link modules installed in the Q2ASCPU basic base module and the extension base module, then select Execute (Y).



Press the Esc key to register the set data.

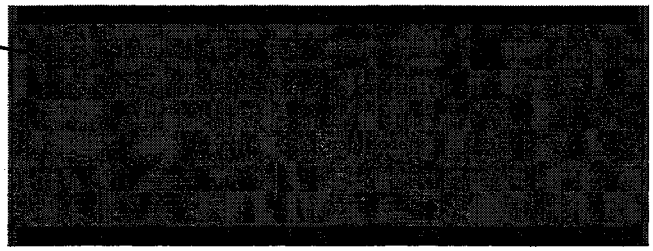
(c) Move the cursor to the module number to perform automatic refresh setting, then press the F3 (details) key to display the "CC-Link Setting window."
Select "4. Auxiliary setting" or "5. Station information setting" to perform detailed data setting.


Set the CC-Link module I/O number for the specified module number.
Set the CC-Link module type for the specified module number.
Set the Q2ASCPU device to use.
Set the remote station connected to the CC-Link module (See (D).)
Set each remote station (See (E).)

Select Execute (Y) or press the Esc key to return to the CC-Link setting list window.

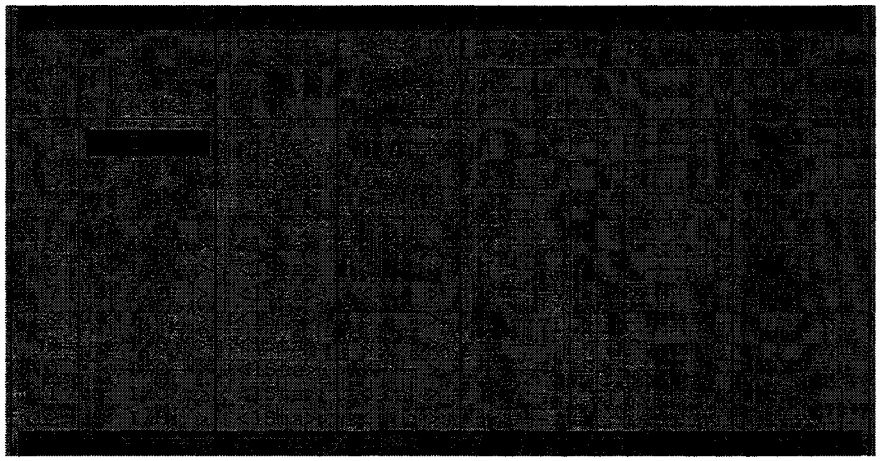
1) Select "4. Auxiliary setting" to display the "Auxiliary Setting window."
Press the Esc key to return to the CC-Link setting window described in step (c).

Set the number of remote stations connected to the CC-Link.



2) Select "5. Station information setting" to display the "Station Information Setting window."
Pressing the Esc key returns to the CC-Link setting window described in step (c).

Set the number of stations specified in step (d).



8 Debug Function

8.1 Function List

There are many convenient functions when using the debugger in Q2ASCPU.
The debug functions are as shown below:

Item	Description	Reference
Monitoring function	Function to read program and device status of the CPU from the GPP function peripheral device	Section 8.2
Writing during run	Function to write in the program while the CPU module is running	Section 8.3
Actual time operation	Function to display the processing time of the program being executed	Section 8.4
Program list monitoring	Function to display the processing time of the programs being executed	Section 8.4.1
Interrupt program list monitoring	Function to display the number of interrupt program executed	Section 8.4.2
Scan time estimate	Function to calculate the intended interval of the program execution time	Section 8.4.3
Sampling trace function	Function to sample the device continuously in the specified timing in the CPU module.	Section 8.5
Status latch function *1	Function to sample the device in the specified moment	Section 8.6
Step operation	Function to execute only the program one step, only a section, or skipping a section	Section 8.7
Step execution	Function to execute the program step by step	Section 8.7.1
Partial execution	Function to execute the specified area of the program	Section 8.7.2
Skip execution	Function to executed the program by skipping the specified area	Section 8.7.3
Program trace function *1	Function to sample the program execution status	Section 8.8
Simulation function *2	Function to simulate the execution by separating the I/O module and special function module	Section 8.9
Multiple request debug function	Function to execute the debugger simultaneously from multiple GPP function peripheral devices	Section 8.10
Monitoring trace function	Function to trace the device continuously in the timing specified in the GPP function peripheral device	—

Refer to GPP function operating manual for details of the operation methods for each function.
*1: When this function is executed, the memory card is required.
*2: When part of this function is executed, the memory card is required.

8.2 Monitoring Function

This is a function to read the program and device status of the CPU module through the GPP function peripheral device.

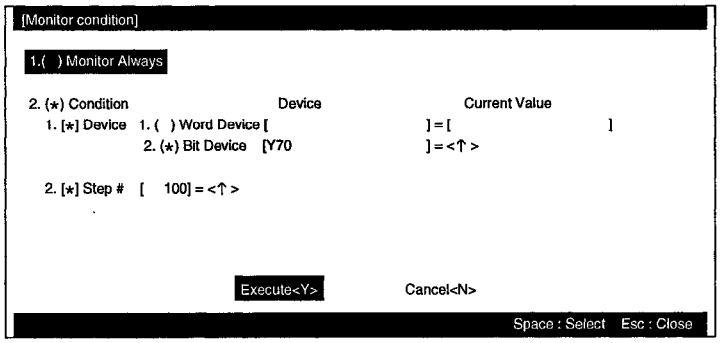
Application

Monitoring conditions to monitor the PC operation status in precise timings can be set. In the "Monitor Condition," the step number to be monitored, continuity status of the step, and the device status can be set, as well as the monitoring can be executed in the END processing. Also, by setting the monitoring stop condition, the precisely timed monitoring screen can be maintained.
When monitoring a Q2ASCPU of function version "B" using SW2IVD/NX-GPPQ, the local device can also be monitored and tested by specifying local device support setting.

8.2.1 Monitor Condition Setting

Function Description

(1) The monitoring conditions can be set.
Each operation is performed in the "Monitor" menu of the circuit mode.
The setup example of the monitor condition is shown below:



The following is the description of the screen:
"1. ( ) Monitor Always" or "2. ( ) Condition" can be set as the monitor condition.

(a) When "1. ( ) Monitor Always" is selected:
The monitor data sampling timing is at every scan after the PC CPU END processing.

(b) When "2. ( ) Condition" is selected:
"1. [ ] Device" or "2. [ ] Step #" can be specified.

1) When "2. [ ] Step #" is specified:
The monitor data sampling timing is when the status right before the execution of the specified step becomes "Specify Status".
The specification method for the execution status is indicated below:

(I)	When changing from non-executing state to executing state	:	<↑>
(II)	When changing from executing state to non-executing state:	:	<↓>
(III)	Always only when executing	:	< ON >
(IV)	Always only when not executing:	:	< OFF >
(V)	Always regardless of status	:	< Always >

Remarks

(1) When "Step # [0]" is specified, set the condition to "Always".

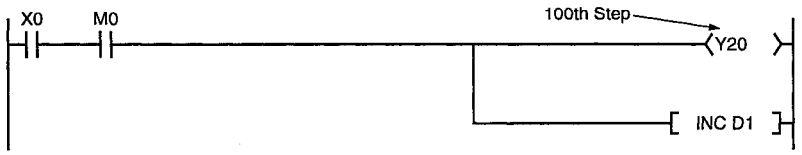
(2) When "1. [ ] Device is specified (when "2. [ ] Step #" is not specified), the monitor data sampling timing is at every scan after the PC CPU END processing. If the data changes in the same scan, it not be detected. (This includes low-speed programs.)

2) When only "1. [ ] Device" is specified
"1. ( ) Word Device" or "2. ( ) Bit Device" can be specified.
a) When "1. ( ) Word Device" is selected:
The monitor data sampling timing is when the current value of the specified word device becomes the specified value in the END processing of the scan.
The specification method for the present value is shown below:
(I) When specified in decimal : <K [Decimal]>
(II) When specified in hexadecimal : <H [Hexadecimal]>
b) When "2. ( ) Bit Device" is specified:
The monitor data sampling timing is when the execution status of the specified bit device becomes the specified state in the END processing of the scan.
The specification method for the execution state is shown below:
(I) When starting : <↑>
(II) When shutting : <↓>

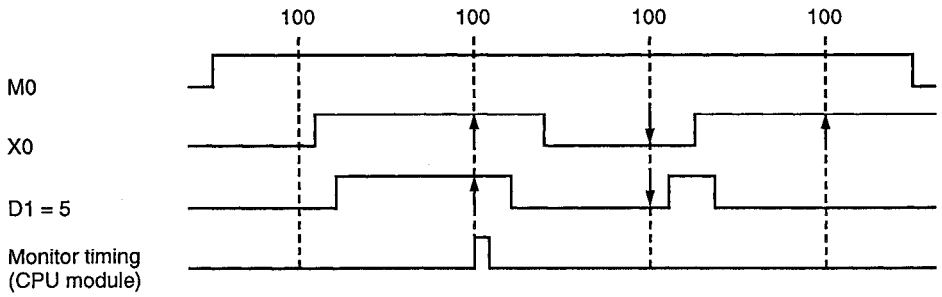
3) When "2. [ ] Step #" or "1. [ ] Device" is specified:
The monitor data sampling timing is when the state right before execution of the specified step or the execution state (present value) of the specified bit device (word device) becomes the specified state.

POINT

When "Step # [100] = <↑>, Word Device [D1]=[K5]" is specified as the detailed condition in the following circuit:



The monitor timing is as shown below. The monitor interval with the GPP function peripheral device depends on the processing speed of the GPP function peripheral device.
The data change shorter than a monitor interval can only be sampled once.



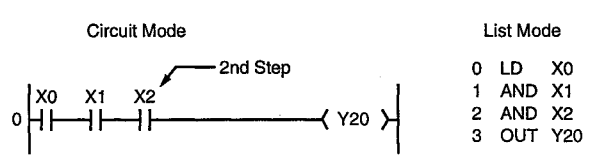
POINTS

(1) When "Step # [2] = <ON>" is specified as the detailed condition in the following circuit:

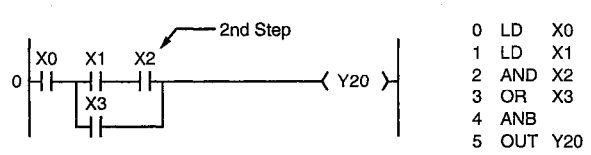
For (1), the monitor execution condition is "X0 and X1 are ON" and for (2), it is "X1 is ON (regardless of whether X0 is ON/OFF)".

When a step during the AND/OR block is specified as the monitor condition, the monitor data sampling timing is when the state right before execution of the specified step from the block LD command becomes the specified state.

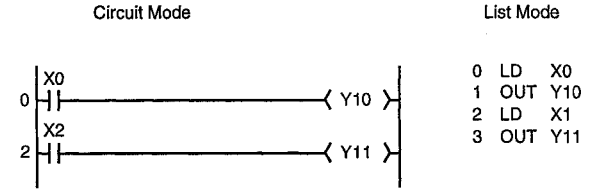
(a)



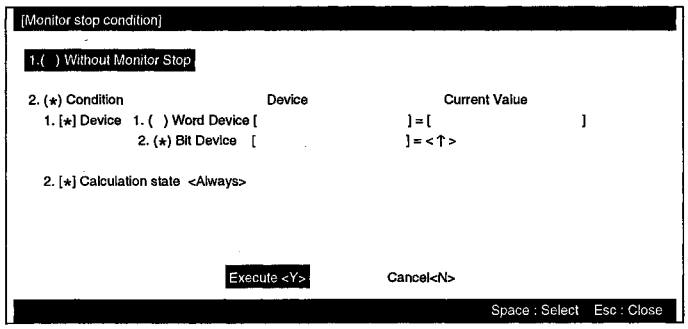
(b)



(2) If the beginning of the circuit block other than step 0 is specified in step No. as the detailed condition, monitor data is collected when the execution status of the instruction right before execution of the specified step becomes the specified status.
If "Step # [2] = <ON>" is specified in the following circuit, monitor data is collected when OUT Y10 turns ON.



(2) The monitor stop condition can be set.
Each operation is performed in "Monitor" screen of the circuit mode.
The setup example of the monitor stop condition is shown below:


[Monitor stop condition]
1.( ) Without Monitor Stop
2. (*) Condition
Device
Current Value
1. [*] Device 1. ( ) Word Device [ ] = [ ]
2. (*) Bit Device [ ] = <↑>
2. [*] Calculation state <Always>
Execute <Y>
Cancel<N>
Space : Select Esc : Close

The following is the description of the screen:
"1 ( ) Without Monitor Stop" or "2. ( ) Condition" can be set as the monitor condition.

(a) When "1. ( ) Without Monitor Stop" is selected:
The monitor is stopped when the [Esc] key is pressed.

(b) When "2. ( ) Condition" is selected:
"1. [ ] Device" or "2. [ ] Calculation State" can be specified.

1) When "2. [ ] Calculation State" is specified:
The monitor stop timing is when the execution state of the step specified in the monitor condition becomes the specified state.
The specification method for the execution status is indicated below:

(I)	When changing from non-executing state to executing state	:	<↑>
(II)	When changing from executing state to non-executing state:	:	<↓>
(III)	Always only when executing	:	< ON>
(IV)	Always only when not executing:	:	< OFF >
(V)	Always regardless of status	:	< Always >

When "2. [ ] Calculation State" is not specified, the monitor stop timing is after the PC CPU END processing.

2) When only "1. [ ] Device" is specified:
"1. ( ) Word Device" or "2. ( ) Bit Device" can be specified.

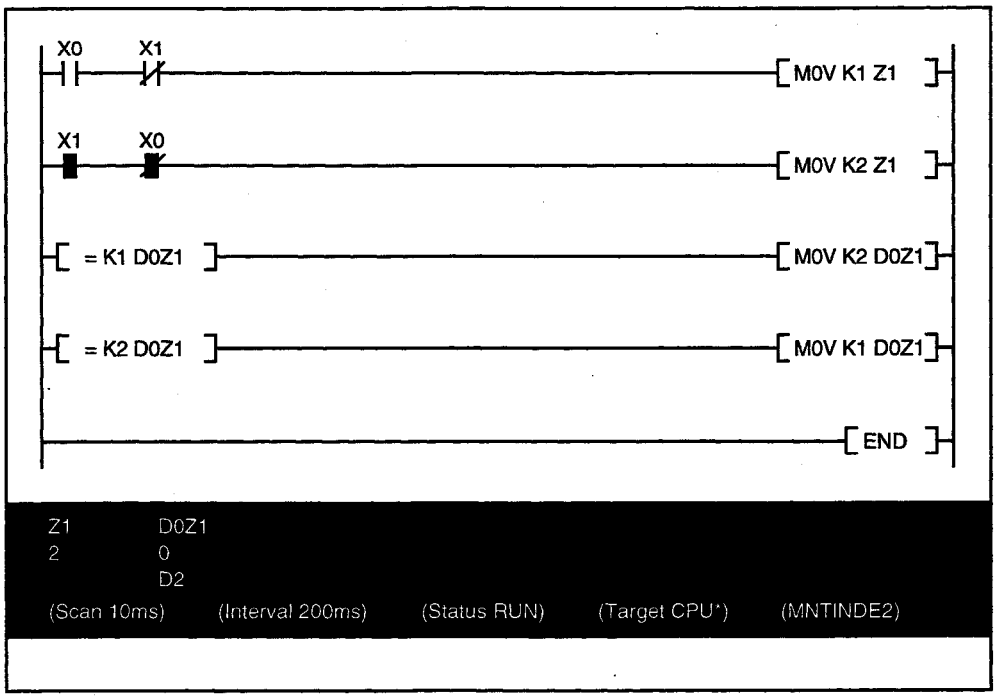
a) When "1. ( ) Word Device" is selected:
The monitor stop timing is when the current value of the specified word device becomes the specified value.
The specification method for the present value is shown below:

(I) When specified in decimal (word) : <K [Decimal]>
(II) When specified in hexadecimal (word) : <H [Hexadecimal]>
(III) When specified in decimal (double word) : <K [Decimal] [Space] [L]>
(IV) When specified in hexadecimal (double word) : <H [Hexadecimal] [Space] [L]>
(V) When actual value is specified : <E [Actual Value]>

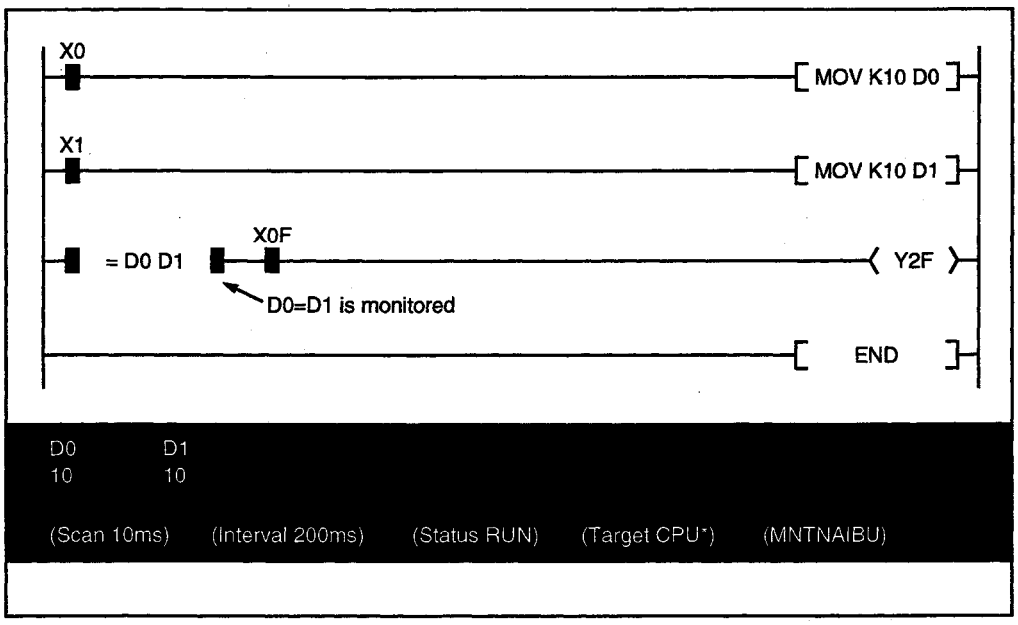
b) When "2. ( ) Bit Device" is specified:
The monitor stop timing is when the execution status of the specified bit device becomes the specified state.
The specification method for the execution state is shown below:

(I) When starting : <↑>
(II) When shutting : <↓>

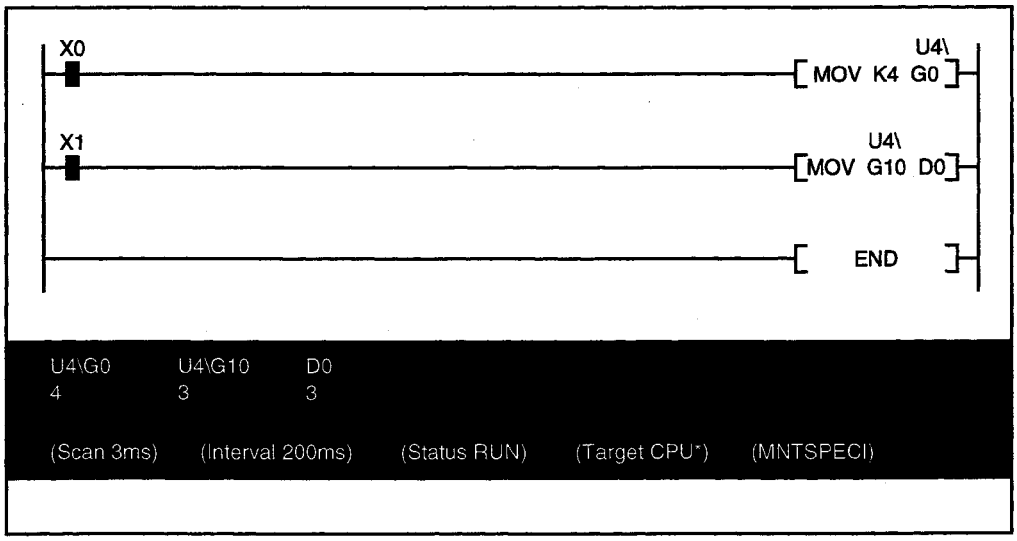
(3) The indexed device can be monitored with the indexed values. The monitoring example is shown below:



(4) The continuity status of the comparison instruction can be monitored.
The monitoring example is shown below:



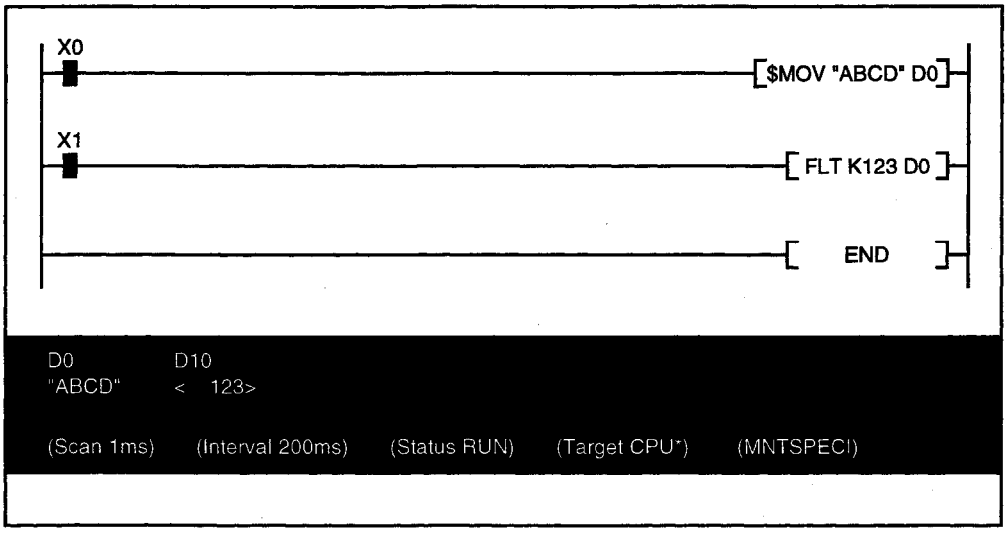
(5) The special function module device can be monitored.
The monitoring example is shown below:



Remarks

When monitoring the special function module device, set "5/Set Monitor" in the "Option" menu of the circuit mode to "1. (*) Monitor".

(6) The actual value and character string can be monitored,
The monitoring example is shown below:



(7) The devices that can be monitored is as follows:

(a) Bit Device : X, FX, DX, Y, FY, DY, M, L , F, SM, V, B, SB,T (contract), T (coil), ST (contact), ST (coil), C (contact),C (coil), J□\XJ□\Y, J□\B, J□\SB, and BL□\S.

(b) Word Device : T (current value), ST (current value), C (current value), D, SD, FD, W, SW, R, Z, ZR, U□\G, J□\W, and J□\SW.

(8) The devices that can be set in the detailed condition is as follows:

(a) Bit Device : X, FX, Y, FY, M, L , F, SM, V, B, SB, T (contact), ST (contact), C (contact), J□\X, J□\Y, J□\B, J□\SB, and BL□\S.

(b) Word Device : T (current value), ST (current value), C (current value), D, SD, FD, W, SW, R, Z, ZR, U□\G, J□\W, and J□\SW.

The following qualification can be set for the above devices:

- Bit device number of digits specification
- Word device bit number specification

Precaution

(1) When monitoring after setting the monitor condition, the file displayed on the GPP function is monitored. Match the file to be monitored by executing the "New PC Read" and file name on the GPPQ.

(2) When monitoring the special function module buffer memory with a direct device specification, and the special function module is malfunctioning or does not exist, FFFFH is monitored.

(3) When monitoring the file register and the file register is not specified, FFFFH is monitored.

(4) Perform the monitoring by matching the device allocation of the CPU module and GPP function.

(5) When the local device in each program file is monitored, the monitoring operation changes depending on whether there is the function version "B" of the Q2ASCPU or not and the type of the GPP function.

[When function version "B" is not described]

- Set the detailed conditions (step No. and device condition) to each program file and perform monitoring.

[When function version "B" is described]

- When GX Developer or SW2IVD/NX-GPPQ is used, setting local device compatibility allows the local device to be monitored in each program file. (Refer to Section 8.2.2.)
- When using SW0IVD/NX-GPPQ or SW1IVD/NX-GPPQ, monitor the local device by performing the same operation as when function version "B" is not described.

(6) When monitoring the buffer memory of the special function module, the scan time takes a longer time just like when executing the FROM/TO instruction.

(7) Multiple users can execute monitoring at the same time.
When multiple users are executing monitoring at the same time, take Note of the following:
High speed monitoring can be performed by increasing 1k step in the system area for every stations monitor file during formatting of internal memory.
Up to 15 stations can be set as the station monitor file, but the program space will be reduced.
The detailed condition setting for the monitoring can only be set for one user.

(8) The monitoring detailed condition setting can only be set in circuit monitor.

(9) When the same device is specified as the monitor condition and monitor stop condition, specify "on" or "off".

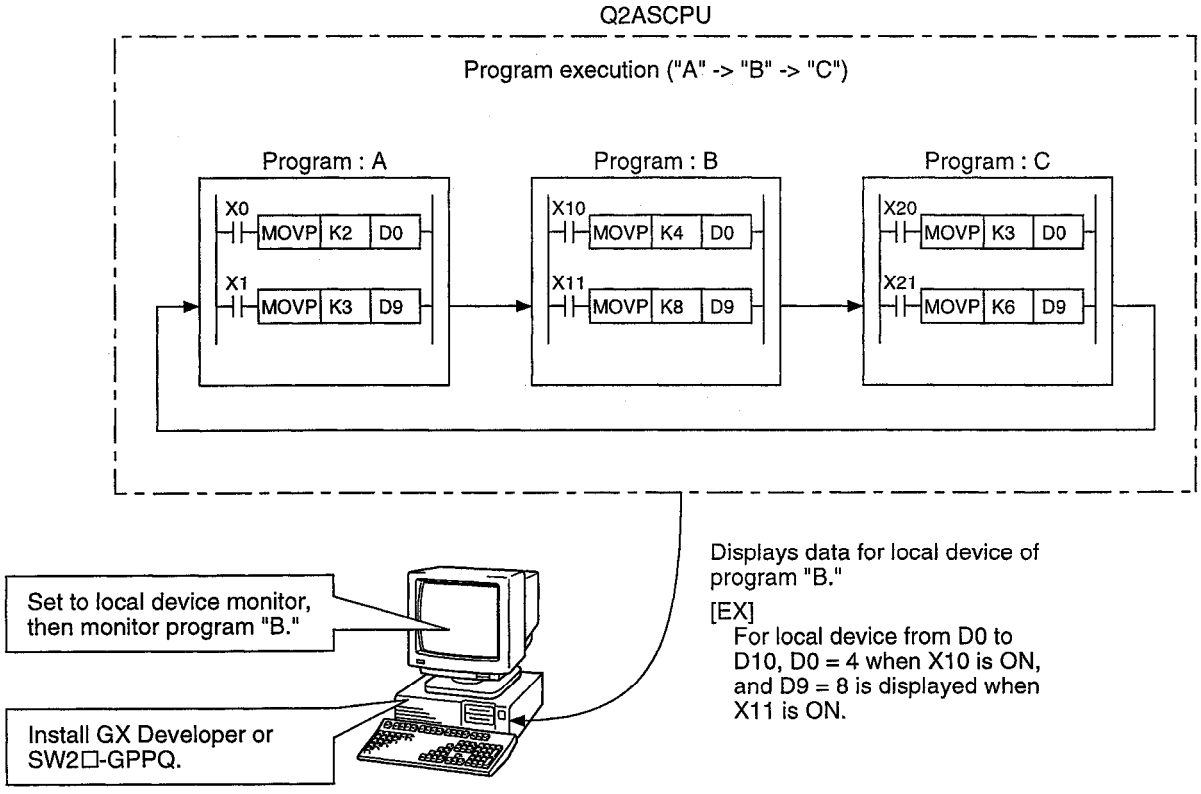
(10) If the step No. is specified as the monitor conditions, the monitor condition will not be met when the instruction at the corresponding step is not executed as described below.
a) When the instruction at the corresponding step is skipped using the CJ, SCJ or JMP instruction.
b) When the corresponding instruction is an END instruction and an FEND instruction exists in the program.

(11) Do not reset the CPU module during monitor condition registration.

8.2.2 Monitoring Test for Local Device

The device set as a local device using the "parameter device setting" can be monitored and tested from a peripheral device.
This function enables debugging while checking the contents of local devices used by programs that are monitored by peripheral devices.

Perform a local device monitor setting for the Q2ASCPU in which programs A, B, and C are being run. Then display program B to monitor its local devices.


Q2ASCPU
Program execution ("A" -> "B" -> "C")
Program : A
X0
MOVP K2 D0
X1
MOVP K3 D9
Program : B
X10
MOVP K4 D0
X11
MOVP K8 D9
Program : C
X20
MOVP K3 D0
X21
MOVP K6 D9
Displays data for local device of program "B."
[EX]
For local device from D0 to D10, D0 = 4 when X10 is ON, and D9 = 8 is displayed when X11 is ON.
Set to local device monitor, then monitor program "B."
Install GX Developer or SW2□-GPPQ.

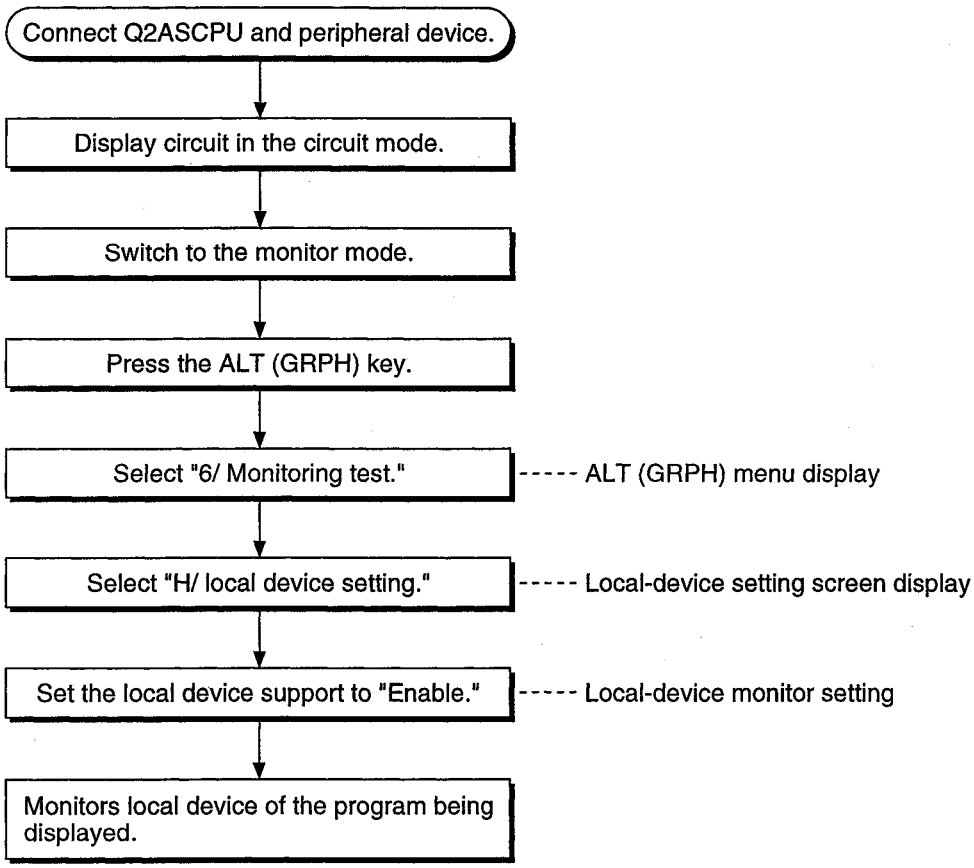
(1) Peripheral device

Monitoring or testing a local device requires the GPP function software packages listed below.

- IBM PC/AT or 100% compatible : SW2IVD-GPPQ type GPP function software package, GX Developer

(2) Monitoring procedure for a local device

Use the following procedure to monitor a local device.


Connect Q2ASCPU and peripheral device.
Display circuit in the circuit mode.
Switch to the monitor mode.
Press the ALT (GRPH) key.
Select "6/ Monitoring test."
----- ALT (GRPH) menu display
Select "H/ local device setting."
----- Local-device setting screen display
Set the local device support to "Enable."
----- Local-device monitor setting
Monitors local device of the program being displayed.

(3) Q2ASCPU and different GPPQ operations by versions

Table 8.1 explains the operations when a local device is set to D0 to D99 and three programs ("A," "B," and "C") are being run on Q2ASCPU.
(Sequence of program execution is A → B → C → (END processing) A → B, and so forth.)

Table 8.1 The Q2ASCPU function version and different GPPQ operations

		Monitor device			
		Function version "B" recorded		Function version "B" not recorded	
		D0	D100	D0	D100
SW0a-GPPQ SW1a-GPPQ		Monitors D0 of program "C."	Monitors D100 after program "C" execution.	Monitors D0 of program "C."	Monitors D100 after program "C" execution.
SW2a-GPPQ	Local device with no setting	Monitors D0 of program "C."	Monitors D100 after program "C" execution.	Monitors D0 of program "C."	Monitors D100 after program "C" execution.
	Local device with setting	Monitors D0 of the program being displayed.	Monitors D0 after executing the program being displayed.	Error (error code: 4001)	

Remarks

GX Developer support functions provided by the function version "B."

(4) Precautions

(a) Each peripheral device can monitor or test the local devices of only one program.

(b) Monitoring or testing cannot be performed simultaneously on local devices of multiple programs from a single peripheral device.

(c) Local devices of up to 16 programs can simultaneously be monitored or tested from multiple peripheral devices.

(d) Monitoring local devices for a standby program reads and stores the local device data; thus the scanning time is extended as shown below:

- Q2ASCPU(S1) : 560 + 1.3 * (the number of words in a local device) [μs]
- Q2ASHCPU(S1) : 220 + 0.8 * (the number of words in a local device) [μs]

8.3 Writing During RUN

This is a function to write in a program during CPU RUN.

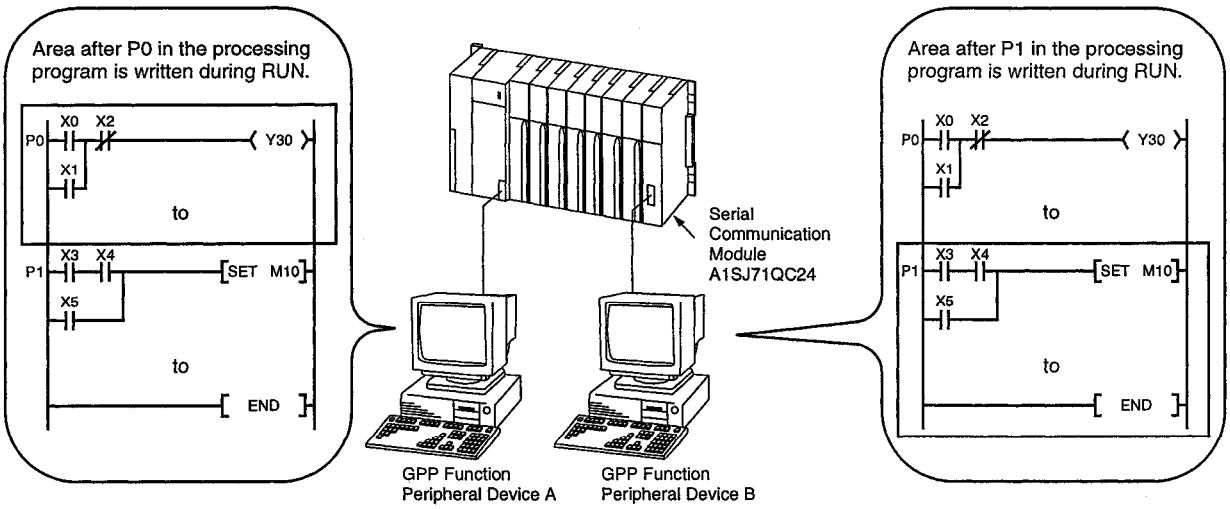
⚠ CAUTION
- Before making program modification during operation, read the manual carefully and ensure safety completely.
  Any operation mistake made while writing during RUN could cause machine damage or accidents.

Application

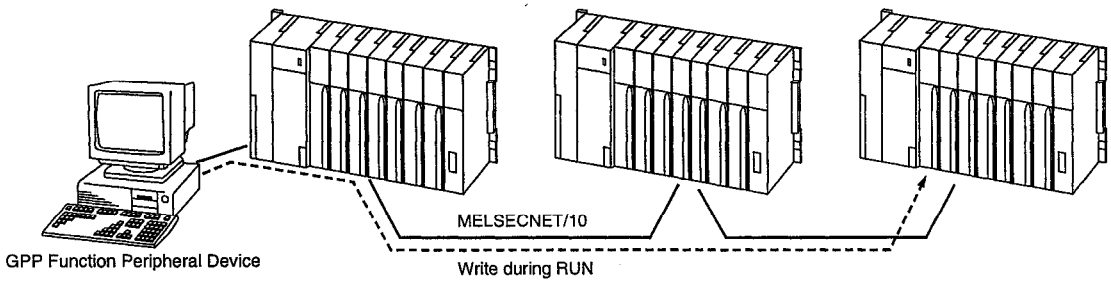
This is used when changing the program without stopping the program.

Function Description

(1) Writing can be performed for one file from multiple GPP function peripheral device.
Specify the pointer to the program to write from each GPP function peripheral device. By doing so, writing during RUN can be performed from each GPP function peripheral device safely.
The following shows an example of GPP function peripheral device A performing a write during RUN from P0 and GPP function peripheral device B performing a write during RUN from P1. The program surrounded in a square is the area to be written during RUN.



(2) Writing to the program during RUN can be performed from a GPP function peripheral device connected to another station in the network.



POINT

Writing can not be performed during RUN while in the step RUN mode.

Operation Procedure

Writing during RUN may be performed from the GPP function peripheral device in either of the following two ways.

(1) When making circuit conversion after circuit creation in the circuit mode, hold down the [shift] key and press the [F4] key to execute writing during RUN.

(2) Set "4. Write During Run Setting" and "7. Write Method at Write During Run" in "4/Write & Conversion Setting" in the "8/Option" menu of the circuit mode.
When making circuit conversion after circuit creation, press the [F4] key to execute writing during RUN.
A setting example is given below.

[Write & Conversion Setting]	
4. Write During Run Setting	1. (*) Write into PC during Run state. 2. ( ) Write into PC in Stop state. 3. ( ) Don't Write into PC.
7. Write Method at Write During Run	1. (*) Normal 2. ( ) Relatively using Pointer

(a) Select "1. ( ) Write into PC during Run state." for "4. Write During Run Setting".

(b) Select "1. ( ) Normal" or "2.( ) Relatively using Pointer" for "7. Write Method at Write During Run".

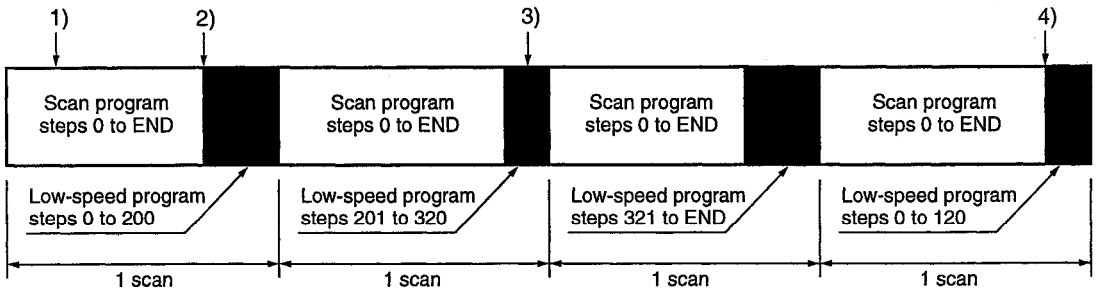
Precaution

Take a Note of the following when writing during RUN:

(1) The memory that can be written during RUN is only internal memory. When writing during RUN is performed during booting, please write to the memory card when stopped.
When booting is started again without performing a write to the memory card, the program without the write during RUN is executed.

(2) A maximum of 512 steps can be written at once during RUN.

(3) When a low-speed program is being executed, the RUN write is started once the low-speed program is complete. Also, the low-speed execution is stopped temporarily during a RUN write.



1): RUN write command of the scan program
2): RUN write execution of the scan program
3): RUN write command of the low-speed program
4): RUN write execution of the low-speed program

8.4 Execution Time Measurement

This is a function to display the processing time of the program being executed.

Application

This is used to find out the effects of each programs processing time to the total scan time.

Function Description

There are three functions to the execution time measurement. The details of each function are indicated in sections 8.4.1 to 8.4.3.

- Program list monitor
- Interrupt program list monitor
- Scan time operation

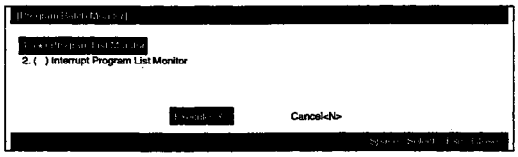
8.4.1 Program List Monitor

This is a function to display the processing time of the program being executed.

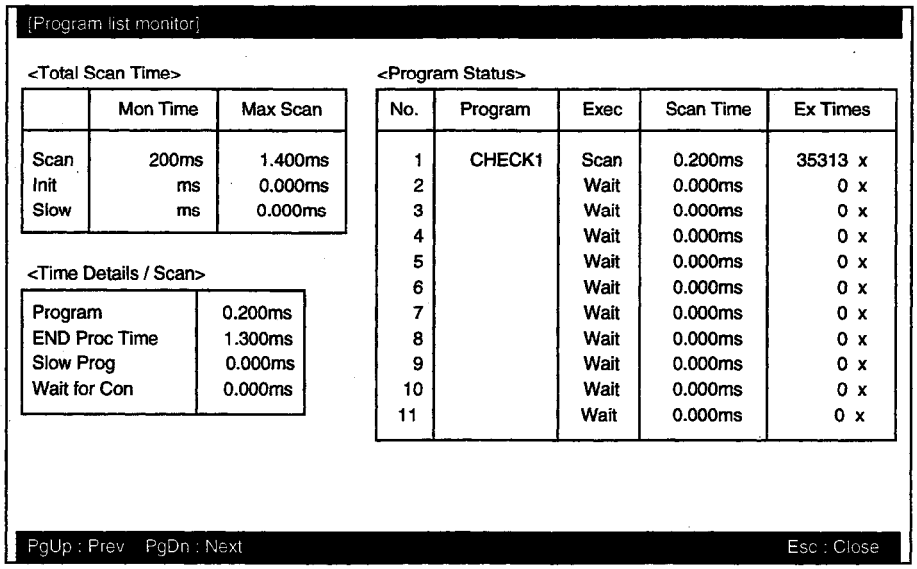
Function Description

The scan time, number of times executed, and processing time by item can be displayed for each program. Each operation is performed in the "Monitor" menu in the monitor status of the circuit mode.

(1) Select "Program Batch Monitor".



(2) Select "Program List Monitor".
An execution example of the program list monitor when the constant scan is set at 120 ms is shown:



<Total Scan Time>

	Mon Time	Max Scan
Scan	200ms	1.400ms
Init	ms	0.000ms
Slow	ms	0.000ms

<Time Details / Scan>

Program	0.200ms
END Proc Time	1.300ms
Slow Prog	0.000ms
Wait for Con	0.000ms

<Program Status>

No.	Program	Exec	Scan Time	Ex Times
1	CHECK1	Scan	0.200ms	35313 x
2		Wait	0.000ms	0 x
3		Wait	0.000ms	0 x
4		Wait	0.000ms	0 x
5		Wait	0.000ms	0 x
6		Wait	0.000ms	0 x
7		Wait	0.000ms	0 x
8		Wait	0.000ms	0 x
9		Wait	0.000ms	0 x
10		Wait	0.000ms	0 x
11		Wait	0.000ms	0 x

The following describes the screen:

(a) "Total Scan Time"
The monitor time set in "5. ( ) PC RAS Setting" in the parameter mode and total scan time for each program type are displayed.

1) "Mon Time"
The monitoring time for the scan program, initialization program, and low-speed program are displayed.
If the scan time exceeds this time, the CPU displays the watchdog timer error.

2) "Scan Time Measurement"
The total time in each item stated in "Scan Time Details for Scan Execution" are displayed.

(b) "Scan Time Details for Scan Execution"
The details of the scan time are displayed.

1) "Program"
The total execution time of the scan program is displayed.

2) "END Processing Time"
The END processing time is displayed.

3) "Low-Speed Program"
The total execution time of the low-speed program is displayed when setting the low-speed program execution time.

4) "Constant Wait"
The constant scan stand by is displayed when setting the constant scan. However, when the low-speed program execution time is set as well, this value is 0.000 ms.

(c) "Program Execution Status"
The program execution status of the program specified in "9. ( ) Auxiliary Setting" of the parameter mode is displayed.

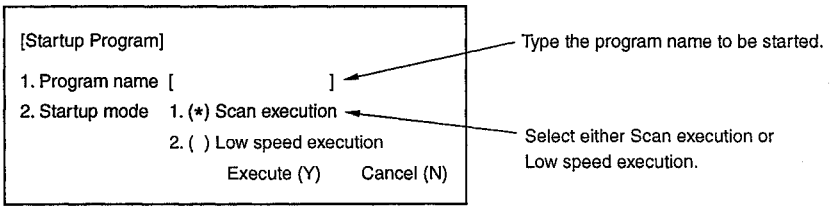
1) "Program"
The program name is displayed in the order set in the parameter.

2) "Exec"
The program type set in the parameter is displayed.

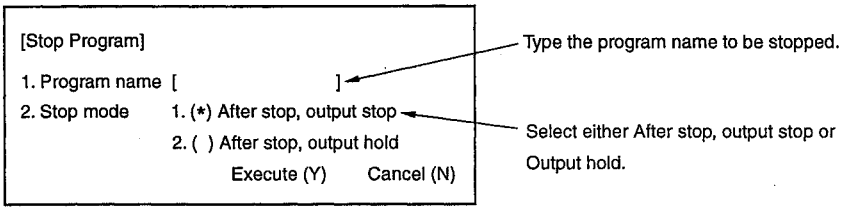
3) "Scan Time"
The actual scan time (current value) is displayed. At the program stop (wait) state, the scan time is displayed as 0.000 ms.

4) "Ex Times"
The number of times the program was executed is displayed, setting the starting point of when the measurement is started as "0". (When the number reaches 65536 times, it is reset to 0.) The "number of executed time" remains even when the program is stopped.

(3) A program start ([F1]) and stop ([F2]) can be made on the program list monitor screen.
(a) Program start ([F1])
A standby program can be set for "Scan execution" or "Low speed execution".
Pressing the [F1] key displays the following window.


[Startup Program]
1. Program name [ ]
2. Startup mode 1. (*) Scan execution
2. ( ) Low speed execution
Execute (Y) Cancel (N)
Type the program name to be started.
Select either Scan execution or Low speed execution.

(b) Program stop ([F2])


[Stop Program]
1. Program name [ ]
2. Stop mode 1. (*) After stop, output stop
2. ( ) After stop, output hold
Execute (Y) Cancel (N)
Type the program name to be stopped.
Select either After stop, output stop or Output hold.

1) If a stop is executed with the stop mode set for "After stop, output stop", the program stops after execution of 1 scan OFF. (Operation is the same as when the POFF instruction is executed.)
When the above stop is executed for a standby program, 1 scan OFF is executed as scan execution and the program then stops.
Hence, "Ex Times" is also incremented by 1.

2) "Ex Times" is also incremented by 1 if the RET/IRET instruction results in an error during execution of 1 scan OFF in the standby program.
At this time, the execution type is "Scan execution".

8.4.2 Interrupt Program List Monitor

This is a function to display the number of executions of the interrupt program.

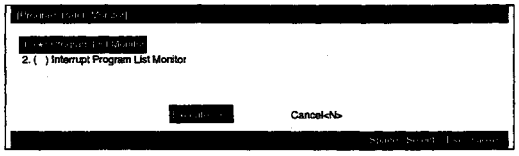
Application

This is used when confirming the execution status of the interrupt program.

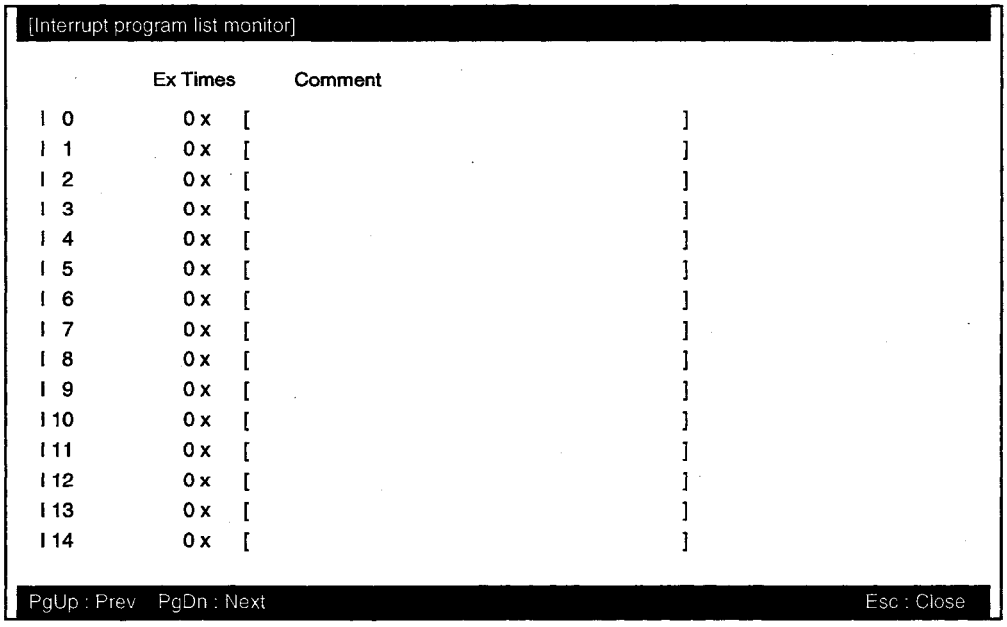
Function Description

The number of executions of the interrupt program can be displayed. Each operation is performed in the "Monitor/Test" menu in the circuit mode.

(1) Select "Program Batch Monitor".



(2) Select "Interrupt Program List Monitor".
The following shows an execution example of the interrupt program list monitor:


[Interrupt program list monitor]
Ex Times
Comment
I 0 0 x [ ]
I 1 0 x [ ]
I 2 0 x [ ]
I 3 0 x [ ]
I 4 0 x [ ]
I 5 0 x [ ]
I 6 0 x [ ]
I 7 0 x [ ]
I 8 0 x [ ]
I 9 0 x [ ]
I 10 0 x [ ]
I 11 0 x [ ]
I 12 0 x [ ]
I 13 0 x [ ]
I 14 0 x [ ]
PgUp : Prev PgDn : Next
Esc : Close

The following describes the screen:

(a) "Ex Times"
The number of times the program was executed is displayed, setting the starting point of when the measurement is started as "0". (When the number reaches 65536 times, it is reset to 0.)
Also, when the operation status is at RUN, this value gets cleared to "0".

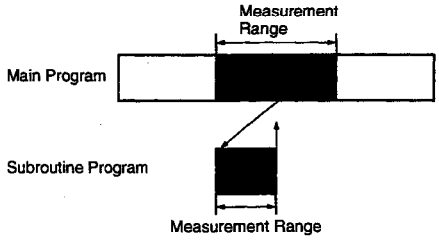
(b) "Comment"
The comment created in the document editor mode is displayed.

8.4.3 Scan Time Measurement

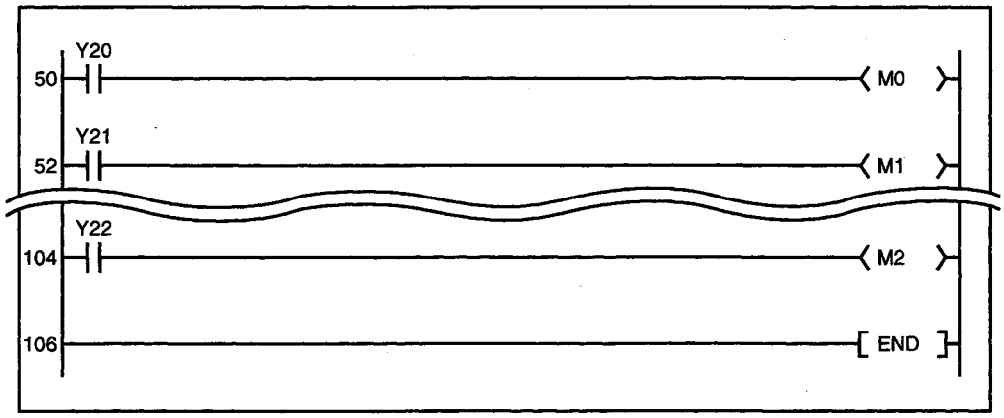
This is a function to display the set program interval processing time.

Function Description

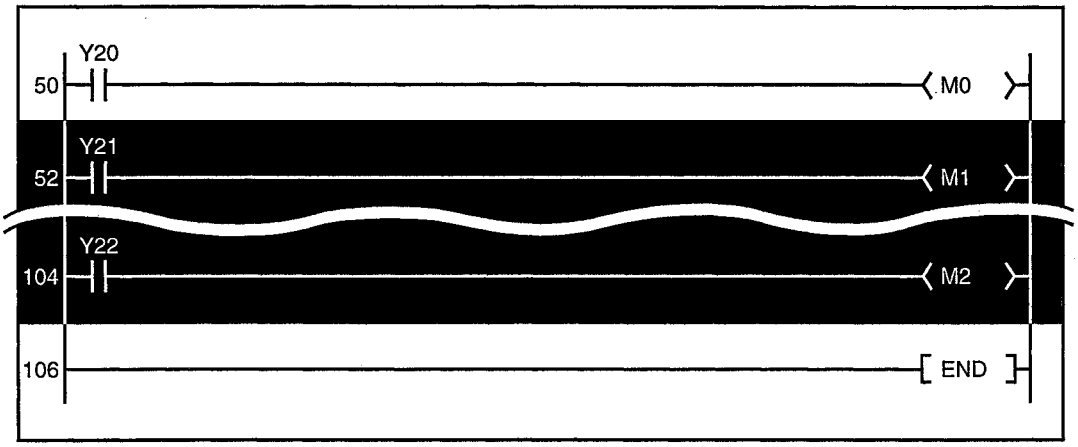
The set interval execution time in the program file can be measured.
The time for the subroutines and interrupt program can be measured as well. If an interrupt program exists, the time includes time taken for the interrupt program.



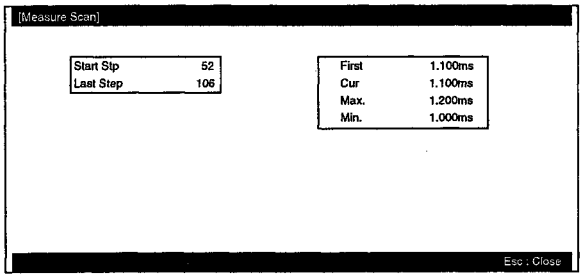
Each operation is performed in the "Monitor" menu of the circuit mode.
(1) Select "Measuring Scan Time".



(2) The scan time measurement range is specified. (The specified area is highlighted.)



(3) The scan time measurement results are displayed.



Precaution

(1) Set the values to be the starting step < complete step.
(2) The time to skip to another program file cannot be measured.
(3) If the measurement time is less than 0.100 ms, 0.000 ms is displayed.

8.5 Sampling Trace Function

This is a function to sample the device continuously on the CPU module at specified timings.

POINT

The memory card is required when executing the sampling trace function.

Application

The changes in the details of the device which uses the program during debugging can be checked in the specified timing.
The debugging time can be minimized using this function.

Function Description

(1) Function

(a) The sampling trace samples the details of the specified device in a set interval (sampling cycle), and stores the trace results to the sampling trace file in the memory card.

(b) The devices that can be traced are as follows:

Bit Device : X, FX, DX, Y, FY, DY, M, L , F, SM, V, B, SB, T (contact), T (coil), ST (contact), ST (coil), C (contact), C (coil), J□\X, J□\Y, J□\B, J□\SB, and BL□\S ..... Maximum 50

Word Device : T (current value), ST (current value), C (current value), D, SD, FD, W, SW, R, Z, ZR, U□\G, J□\W, and J□\SW ..... Maximum 50

(c) The sampling trace file stores the trace condition data and trace execution data necessary to perform the sampling trace. When trace is started with the GPP function peripheral device, the trace is performed for the number of times set.

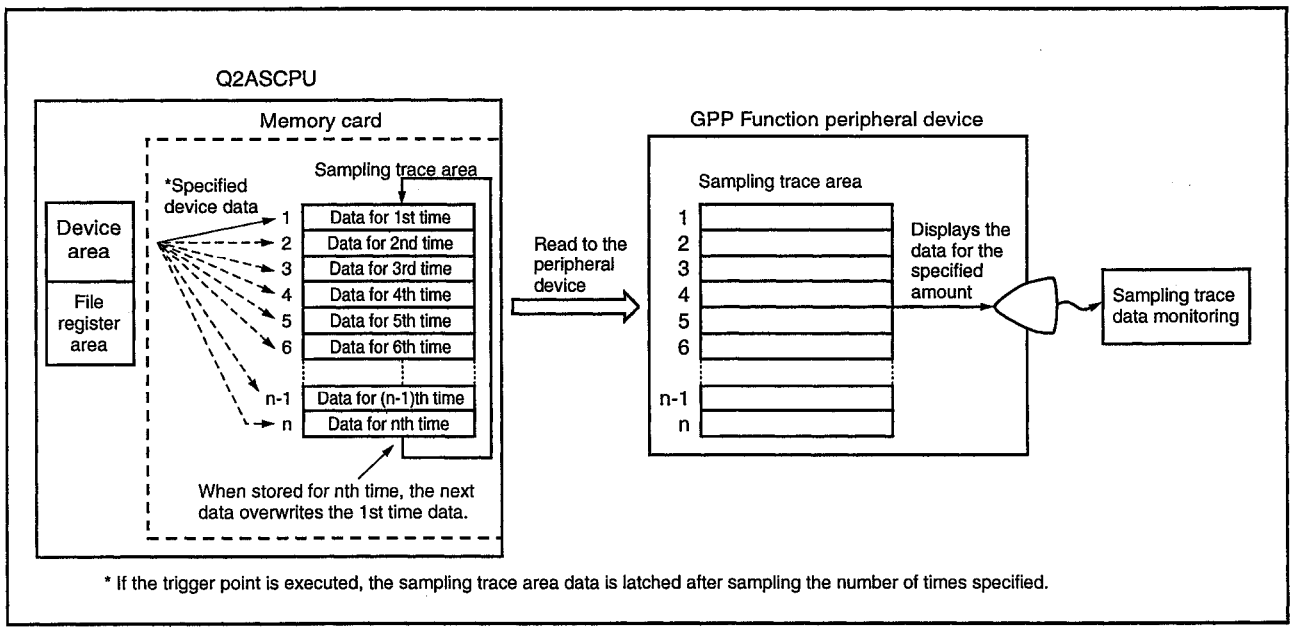
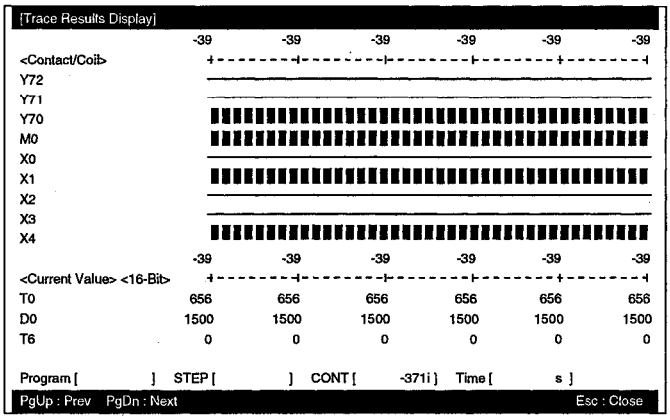

Q2ASCPU
Memory card
Sampling trace area
*Specified device data
Device area
File register area
1 Data for 1st time
2 Data for 2nd time
3 Data for 3rd time
4 Data for 4th time
5 Data for 5th time
6 Data for 6th time
n-1 Data for (n-1)th time
n Data for nth time
When stored for nth time, the next data overwrites the 1st time data.
Read to the peripheral device
GPP Function peripheral device
Sampling trace area
Displays the data for the specified amount
Sampling trace data monitoring
* If the trigger point is executed, the sampling trace area data is latched after sampling the number of times specified.

Fig. 8.1 Sampling Trace Operation

(d) The trace result displays the on/off status of the bit device for the sampling cycle, and the current value of the word device.



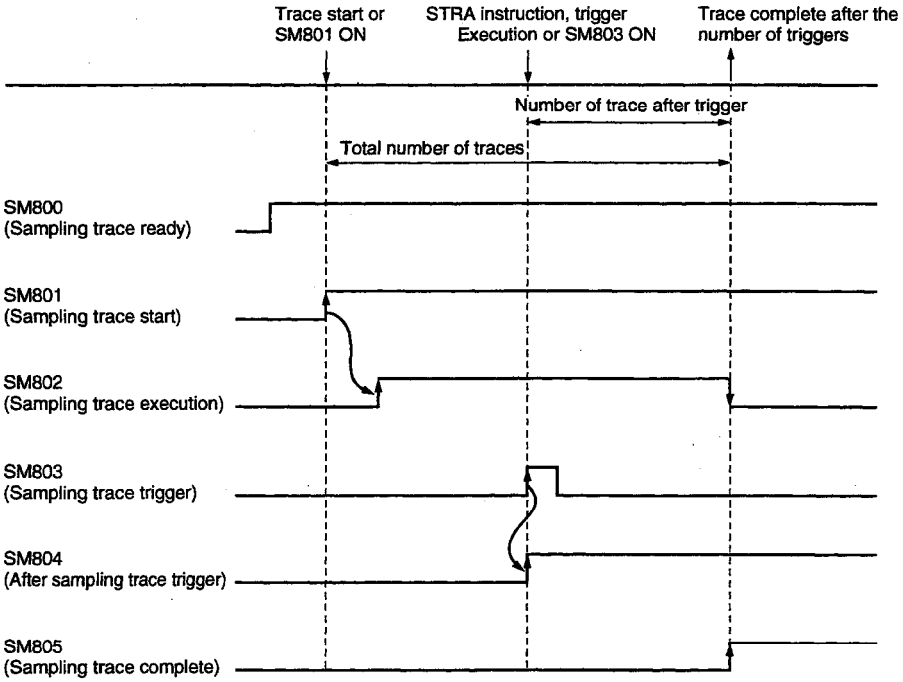
When the CPU is during STOP, the trace result cannot be read since the trace is in a suspended status.

(2) Basic Operation

The basic operation of the sampling trace is as follows:

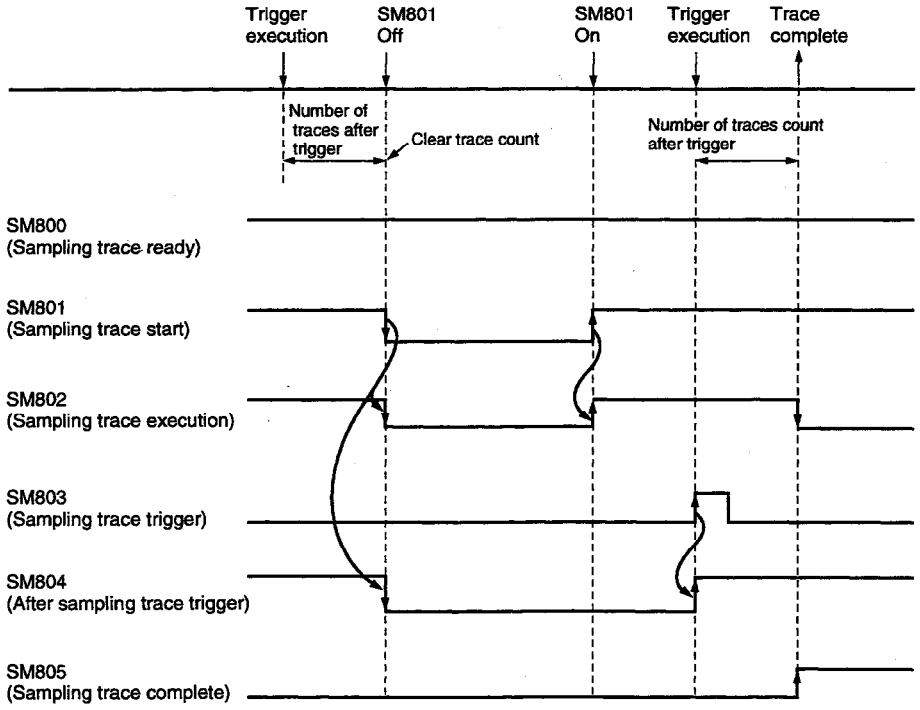
The execution status of the sampling trace function can be checked in the special relay SM800 to SM805 and SM826.

- Trace execution



* SM800 automatically turns on when the sampling trace is ready.

• Trace interrupt



* When trace is interrupted from the GPP function peripheral device, the SM800 is turned off also.

The operation when an error occurs is as follows:

When an error occurs during sampling trace, the SM826 (sampling trace error) is turned on and SM801 (sampling trace start) is turned off.

To turn off SM826, restart the trace.

Operation Procedure

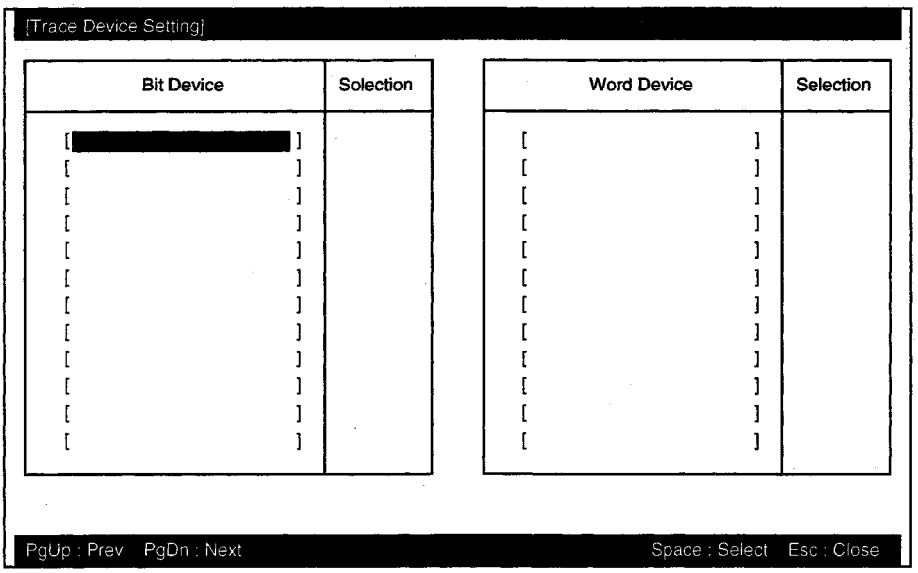
The sampling trace operation is performed in the following manner:

Each operation is performed in the "sampling trace" screen in the online mode trace menu.

(1) Set the trace device and trace condition in GPP function.

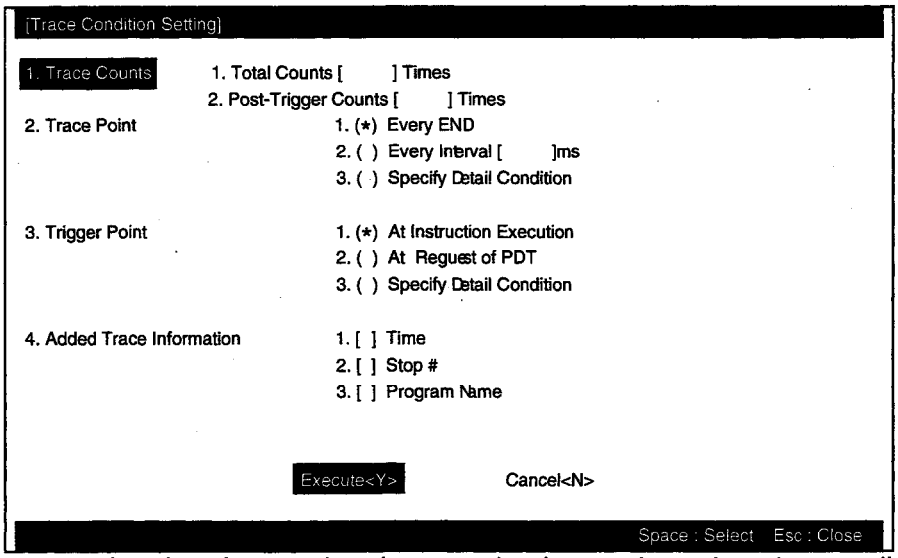
(a) Trace Device Setting

Set the device in the "Trace Device Setting" of the "Sampling Trace" screen.

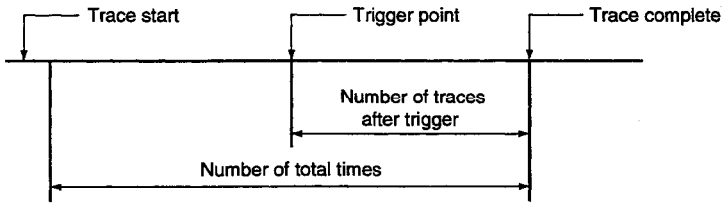


(b) Setting the Trace Condition

Set the device in the "trace condition setting" of the "sampling trace" screen.


[Trace Condition Setting]
1. Trace Counts
1. Total Counts [ ] Times
2. Post-Trigger Counts [ ] Times
2. Trace Point
1. (*) Every END
2. ( ) Every Interval [ ]ms
3. ( ) Specify Detail Condition
3. Trigger Point
1. (*) At Instruction Execution
2. ( ) At Reguest of PDT
3. ( ) Specify Detail Condition
4. Added Trace Information
1. [ ] Time
2. [ ] Stop #
3. [ ] Program Name
Execute<Y>
Cancel<N>
Space : Select Esc : Close

After executing the trigger point, the operation is complete when the sampling is performed for a specified number of times (number of times after trigger).



The following describes the items on the screen.
The trace condition setting can set "1. Trace Counts", "2. Trace Point", "3. Trigger Point", and "4. Added Trace Information".

1) Trace Counts
The trace counts sets the number of times to execute the sampling trace from trace execution to trace complete.
The number of times after trigger sets the number of times to executes the sampling trace from trigger execution to trace complete.
The setting range for each number of times is shown below:
Number of times after trigger ≤ Total number of times ≤ 8192

2) Trace Point Setting
This sets the timing to sample trace data. Select one from the following:

a) Every END : Performed for END instruction for every scan.

b) Set Time : Performed at every specified time. The setting range is between 5 to 5000 ms in 5 ms units.

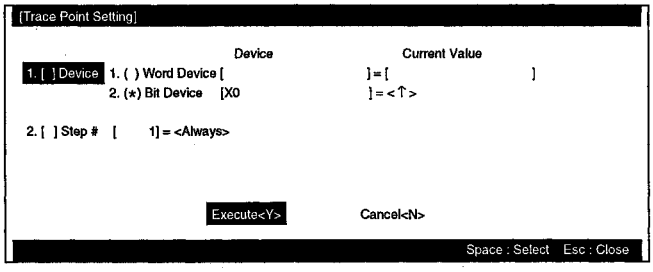
c) Detailed Condition : Sets the device and step no. The setting example is shown below. The setting method and trace data sampling timing is the same as when setting the monitor condition in section 8.2.1 "Monitor Functions".

The devices that can be set in the detailed condition are as follows:

a) Bit Device : X, FX, Y, FY, M, L , F, SM, V, B, SB, T (contact), ST (contact), C (contact), J□\X, J□\Y, J□\B, J□\SB, and BL□\S.

b) Word Device : T (current value), ST (current value), C (current value), D, SD, FD, W, SW, R, Z, ZR, U□\G, J□\W, and J□\SW.

The following attributes can be set for the above devices:
- Bit device number of digits specification
- Word device bit number specification

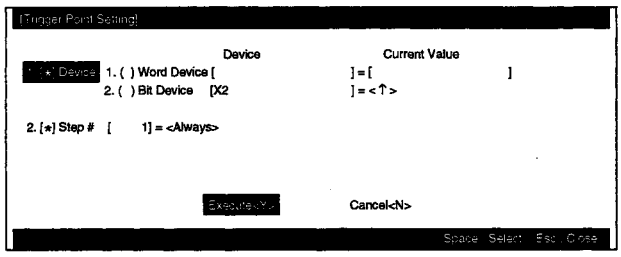


3) Trigger Point Setting
The point to execute the trigger is set. Select one from the following:

a) When instruction is executed : The STRA instruction execution is the trigger.

b) When trigger is operated with peripheral : The trigger operation from a GPP function peripheral device is the trigger.

c) Detailed Condition : The device and step number is set. The setting example is shown below. The setting method and trigger execution timing is the same as the monitor condition setting in Section 8.2, "Monitor Functions".



4) Added Trace Information
The added information for every trace is set. Multiple items can be selected from the following (or none of the items have to be selected):

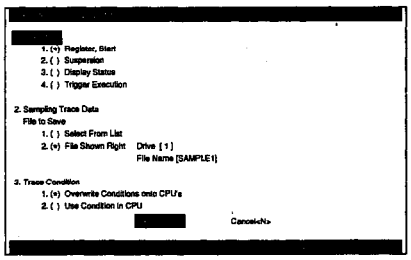
a) Time : Stores the time when the trace was executed.

b) Step Number : Stores the step number when the trace was executed.

c) Program Name : The program name of the program that executed the trace is stored.

(2) The created trace device and trace condition is written to the memory card.

(a) Set the trace file and storage destination.
The drive number and file name are set in the "1. ( ) Execute Trace & Display Status" in the "Sampling Trace" screen.

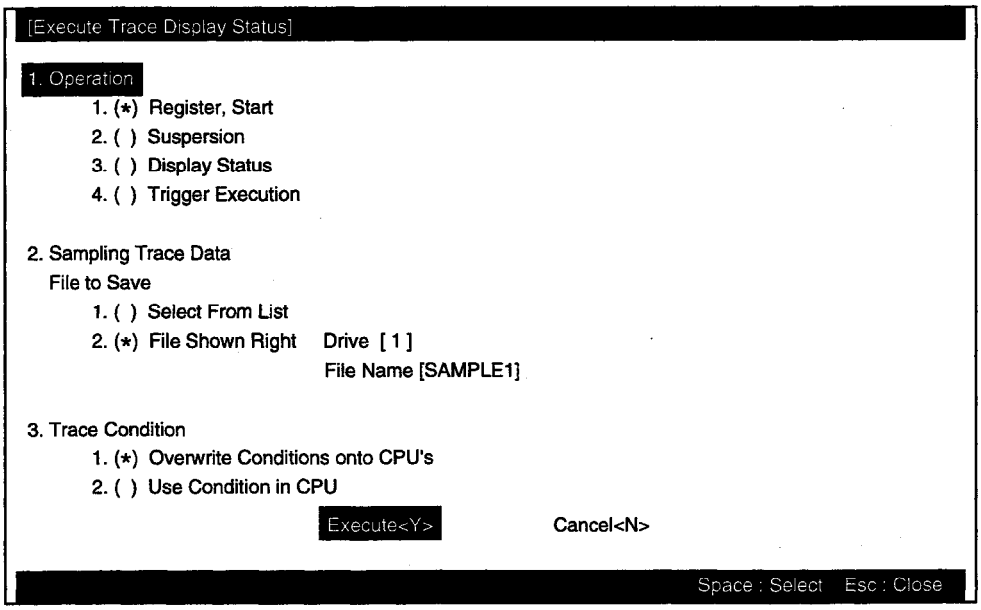

1. (•) Register, Start
2. ( ) Suspension
3. ( ) Display Status
4. ( ) Trigger Execution
2. Sampling Trace Data
File to Save
1. ( ) Select From List
2. (•) File Shown Right Drive [1]
File Name [SAMPLE1]
3. Trace Condition
1. (•) Overwrite Conditions onto CPU's
2. ( ) Use Condition in CPU
Cancel<N>

(b) The trace file is written to the memory card.
The trace file is written to the memory card in "9. ( ) Write to PC (Condition)" in the "Sampling Trace" screen.
The files are written in the memory card as file names, so multiple trace files can be stored.

(3) Sampling trace is executed.

The sampling trace is executed in "1 ( ) Execute Trace & Display Status" in the "Sampling Trace" screen.

The following shows a setting example of "1. ( ) Execute Trace & Display Status".


[Execute Trace Display Status]
1. Operation
1. (*) Register, Start
2. ( ) Suspension
3. ( ) Display Status
4. ( ) Trigger Execution
2. Sampling Trace Data
File to Save
1. ( ) Select From List
2. (*) File Shown Right Drive [1]
File Name [SAMPLE1]
3. Trace Condition
1. (*) Overwrite Conditions onto CPU's
2. ( ) Use Condition in CPU
Execute<Y>
Cancel<N>
Space : Select Esc : Close

The following describes the screen.

"1. Operation", "2. Sampling Trace Data", and "3. Trace Condition" can be set in the execute trace & displays status display.

(a) Operation

Select one from the following:

1) Register Start : The trace is registered and started. Starts the count for trace count.

2) Suspension : The trace is interrupted. The trace count and number of times after trigger are cleared. (When restarting trace, select "Register Start" again.)

3) Display Status : The trace status is displays on the same screen as messages.

4) Trigger Execution : Starts to count the number of times after trigger. The trace is complete when the traces are performed to the number of times after trigger.

(b) Sampling Trace Data

Select one from the following:

1) Selected from List : Selected from the sampling trace files in the memory card.

2) File Shown Right Drive : Sets the drive number and sampling trace file name.

(c) Trace Condition

Select one from the following:

1) Overwrite Conditions Onto CPU's : Overwrites trace condition to the existing trace file.

2) Use Condition in CPU : Executes the program with the conditions in the trace file specified in "2. Sampling Trace Data".

(4) Read the trace results from the CPU module and display the data.
   1) Reads the trace results from the CPU module by "A. ( ) Read from PC (Condition)" in the "Sampling Trace" screen.
   2) With "4. ( ) Trace Results Display" in the "Sampling Trace" screen, the read trace results are displayed.

POINT

When sampling trace is executed once, the trace is not executed second time. When re-executing, execute the STRAR instruction, and reset the sampling trace.

Precaution

(1) Set the sampling trace file in the RAM of the memory card.
(2) The sampling trace can be executed from another station on the network or serial communication module. However, the trace cannot be executed from multiple areas at once. The trace can only be executed from one area with Q2ASCPU.
(3) The trace condition registered in the CPU module is latched. The condition data is stored even if the PLC power is off. To clear data, perform the latch-clear operation with the Q2ASCPU RUN/STOP key switch.
(4) The sampling trace is performed by connecting the Q2ASCPU and GPP function peripheral device.

8.6 Status Latch Function

This is a function to sample data of device for the specified moment.

Application

The device status of the device used by the program is stored in a file when the debug is specified.

Function Description

(1) Function

(a) The status latch stores the device status at the specified moment in the status latch file in the memory card.

(b) The status latch file stores the status latch condition and status latch execution data to perform status latch. The device data is sampled when the SLT instruction is executed in the program, status latch start is executed in the GPP function peripheral device, and according to device and step number condition.

(c) The status latch result displays the ON/OFF status of the bit device at the specified moment, and the word device value.

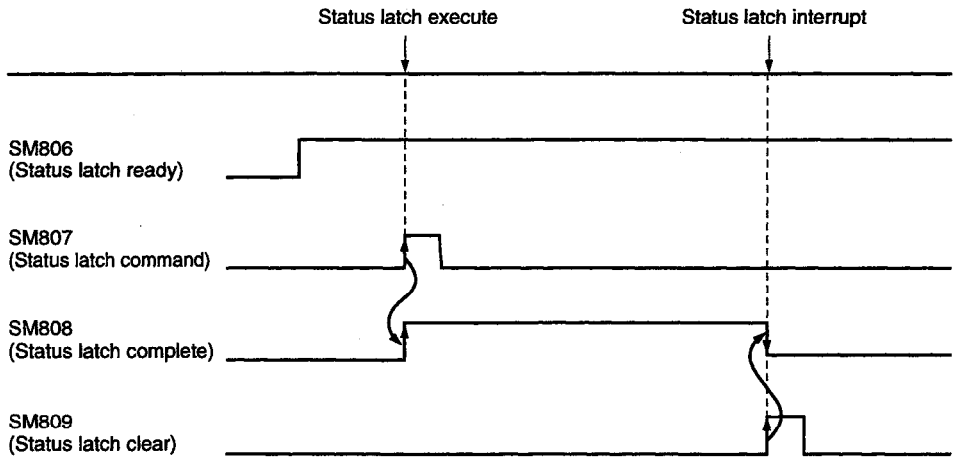
POINT

Memory card is required to execute status latch function.

(2) Basic Operation

The basic operation for status latch is as shown below.

The execution status of the status latch function can be checked in special relay SM806 to SM809, and SM827.



*SM806 automatically turns on when the status latch is ready.

(3) The operation when error occurs is as follows:

When error occurs during status latch, SM827 is turned on and SM808 (Complete) is turned on.

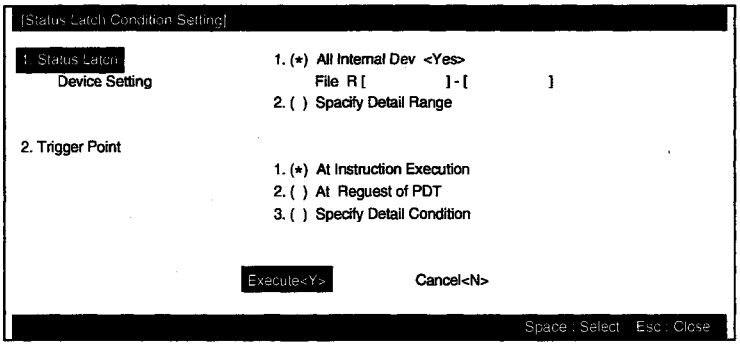
To turn off SM827, turn on SM809 or execute the SLTR instruction.

Operation Procedure

The operation for the status latch is performed in the following manner:
Each operation is performed in the "Status Latch" screen in the online mode trace menu.

(1) Setting the Status Latch Condition

The status latch condition is set in "2. ( ) Status Latch Condition Setting" in the "Status Latch" screen.


[Status Latch Condition Setting]
1. Status Latch
Device Setting
1. (*) All Internal Dev <Yes>
File R[ ]-[ ]
2. ( ) Spacify Detail Range
2. Trigger Point
1. (*) At Instruction Execution
2. ( ) At Reguest of PDT
3. ( ) Specify Detail Condition
Execute<Y>
Cancel<N>
Space : Select Esc : Close

The following describes the screen.

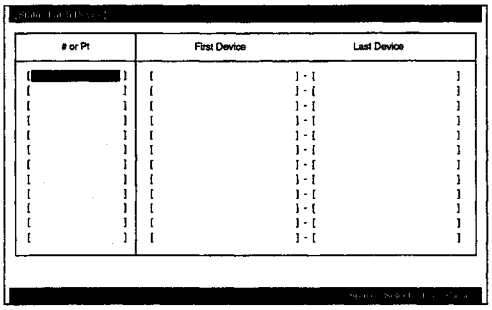
The status latch condition setting can set "1. Status Latch Device Setting" and "2. Trigger Point".

(a) Status Latch Device Setting

The device to execute status latch is set. Select one from the following:

1) All internal devices : Sets whether to latch the all internal devices.

2) Sets the detailed range setting : The device type and points are set. A setting example is shown below:



(Devices that can be used)

1) Bit Device : X, Y, M, L , F, SM, V, B, SB, T (contact), T (coil), C (contact), C (coil), ST (contact), ST (coil), J□\X, J□\Y, J□\B, J□\SB, and BL□\S.

2) Word Device : T (current value), ST (current value), C (current value), D, SD, W, SW, R, ZR, U□\G, J□\W, and J□\SW.

Remarks

Up to 1000 can be set as the device range for bit devices and word devices combined. The qualification of the above devices cannot be performed.

(b) Trigger Point Setting

This sets the condition to execute the status latch. Select one from the following:

1) When executing instruction : The SLT instruction execution is the trigger.

2) When operating the trigger with a peripheral : The trigger operation from the GPP function peripheral device is the trigger.

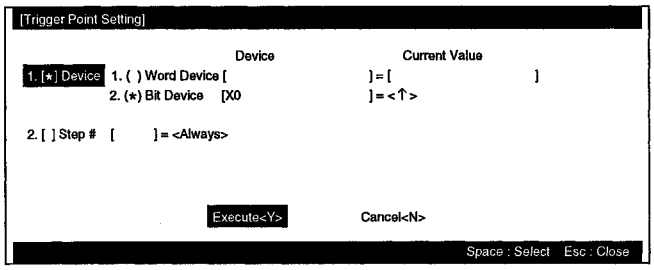
3) Detailed Condition

The device and step number are set.

A setting example is shown below. The setting method and trigger execution timing is the same as the monitor condition setting in Section 8.2.1, "Monitor Functions".

Data collection timing

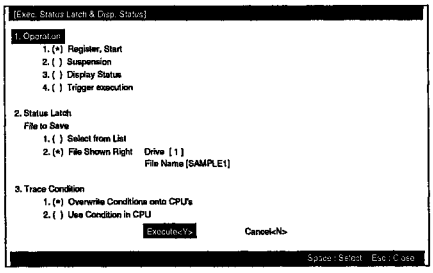
- If only "Device" has been specified, data collection is performed when the trigger condition is met.
- If only "Step #" has been set, data collection is performed at the END processing performed when the trigger condition is met.



(2) The status latch condition created is written into the memory card.

(a) The status latch file and storage destination are set.

The drive number and file name are set in "1. ( ) Exec. Status Latch & Disp. Status" in the "Status Latch" screen.


[Exec. Status Latch & Disp. Status]
1. Operation
1. (*) Register, Start
2. ( ) Suspension
3. ( ) Display Status
4. ( ) Trigger execution
2. Status Latch
File to Save
1. ( ) Select from List
2. (*) File Shown Right
Drive [1]
File Name [SAMPLE1]
3. Trace Condition
1. (*) Overwrite Conditions onto CPU's
2. ( ) Use Condition in CPU
Execute<Y>
Cancel<N>
Space : Select Esc : Close

(b) The status latch file is written into the memory card.

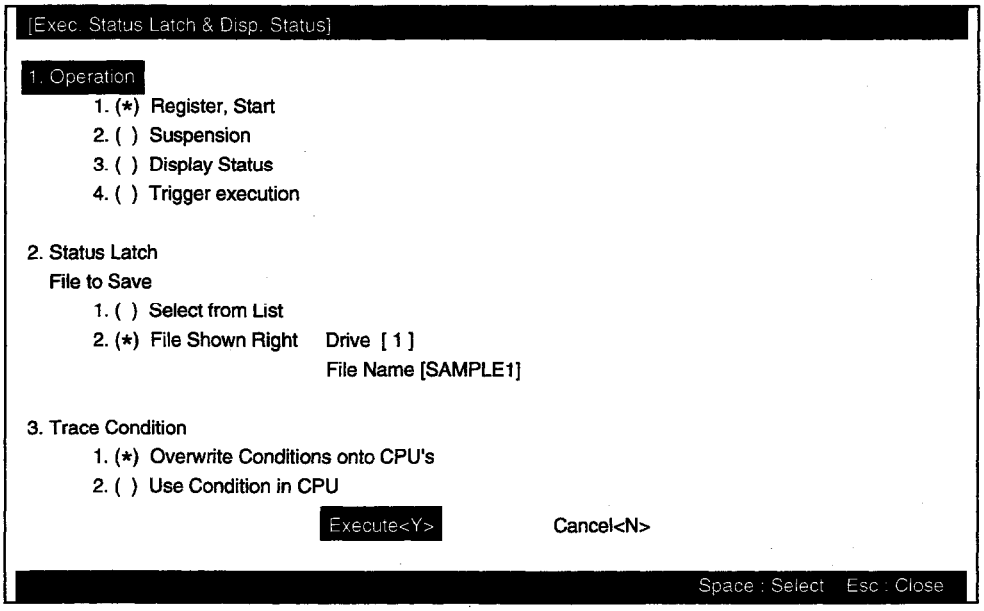
The status latch file is written in the "7. ( ) Write to PC (Condition)" in the "Status Latch" screen.

The file name is written into the memory card, so it can be executed with multiple status latch conditions.

(3) Status latch is executed.

The status latch is executed with "1. ( ) Exec. Status Latch & Disp. Status" in the "Status Latch" screen.

The following shows a setting example of "1. ( ) Exec. Status Latch & Disp. Status".


[Exec. Status Latch & Disp. Status]
1. Operation
1. (*) Register, Start
2. ( ) Suspension
3. ( ) Display Status
4. ( ) Trigger execution
2. Status Latch
File to Save
1. ( ) Select from List
2. (*) File Shown Right Drive [1]
File Name [SAMPLE1]
3. Trace Condition
1. (*) Overwrite Conditions onto CPU's
2. ( ) Use Condition in CPU
Execute<Y>
Cancel<N>
Space : Select Esc : Close

The following describes the screen.

Exec. status latch & disp. status can set "1. Operation", "2. Status Latch", and "3. Trace Condition".

(a) Operation

Select one from the following:

1) Register, Start : The status latch is registered and started. Starts the device data collection.
2) Suspension : The status latch status is cleared.
3) Display Status : The status latch status is displayed on one screen as messages.
4) Trigger Execution : The trigger is executed.

(b) Status Latch

Select one from the following:

1) Select from List : Status latch is selected from the status latch files in the memory card.
2) File Shown Right Drive : The drive number and status latch file name is set.

(c) Status Latch Condition

Select one from the following:

1) Overwrite Conditions onto CPU's : The status latch condition is overwritten to the existing status latch file.
2) Use Condition in CPU : The execution is performed with the conditions in the status latch file specified in "2. Status Latch".

(4) The status latch result is read from the CPU module and displayed.
   (a) The status latch result is read from the CPU module with "8. ( ) Read from PC (Results)" in the "Status Latch" screen.
   (b) The status latch result is displayed by setting "1. Monitor Target" to "3. ( ) Status Latch" in the "Monitor Target Setting" in the circuit mode "Option" menu.

Precaution

(1) Set the status latch file in the memory card RAM.
(2) The status latch can be executed from another station on the network or serial communication module. However, the status latch cannot be executed from multiple areas at once. The status latch can only be executed from one area with Q2ASCPU.
(3) The trace condition registered in the CPU module is latched. The condition data is stored even if the power is off. To clear data, perform the latch-clear operation with the Q2ASCPU RUN/STOP key switch.
(4) The status latch is performed by connecting the Q2ASCPU and GPP function peripheral device.
(5) When the monitor target is set for "Status latch", the timer/counter set values are not displayed.
"0" appears in the timer/counter set value field.

Remarks

When the monitor target is set for "Device memory", the timer/counter set values are not displayed.
"0" appears in the timer/counter set value field.

8.7 Step Drive

This is a function to execute the program step-by-step, execute one part, or execute by skipping one area.

Application

This is used to debug or finding out the cause of an error.

Function Description

This can only be executed when the CPU module is at STEP-RUN.
There are three functions to the step drive. Refer to Sections 8.7.1 to 8.7.3 for descriptions of each function.

- Step Execution
- Partial Execution
- Skip Execution

8.7.1 Step Execution

The step execution executes the operation one step at a time from the specified step of he sequence program operation.
The sequence program can be executed by checking the sequence program execution status or details of each device while debugging.
There are two types of step executions as shown below:

(1) Step-By-Step Execution By Instruction
One instruction is executed from the step that the program is stopped at, and then stops again. This can be used to check the device status for executing each command.

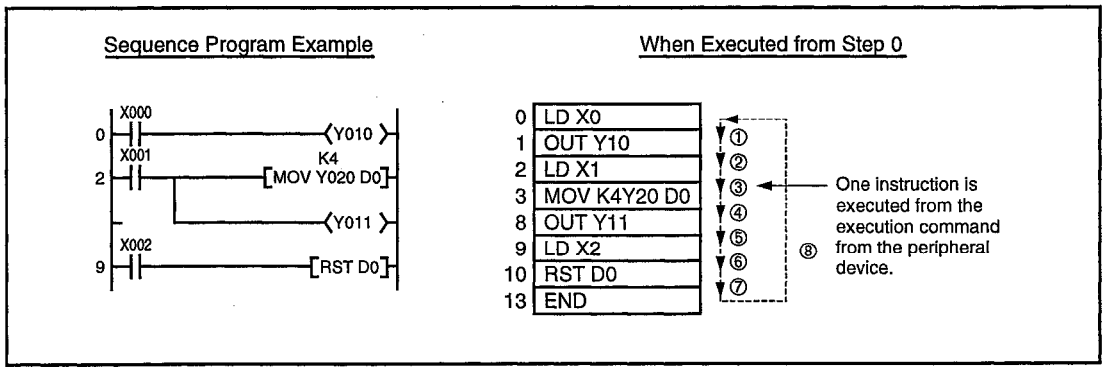


Fig. 8.2 Step-By-Step Execution By Command

(2) Step-By-Step Execution by Number of Repeats Specification
The program is executed for a specified number of loops (1 to 32767) from the presently stopped step or step 0, then stops (at a break point).

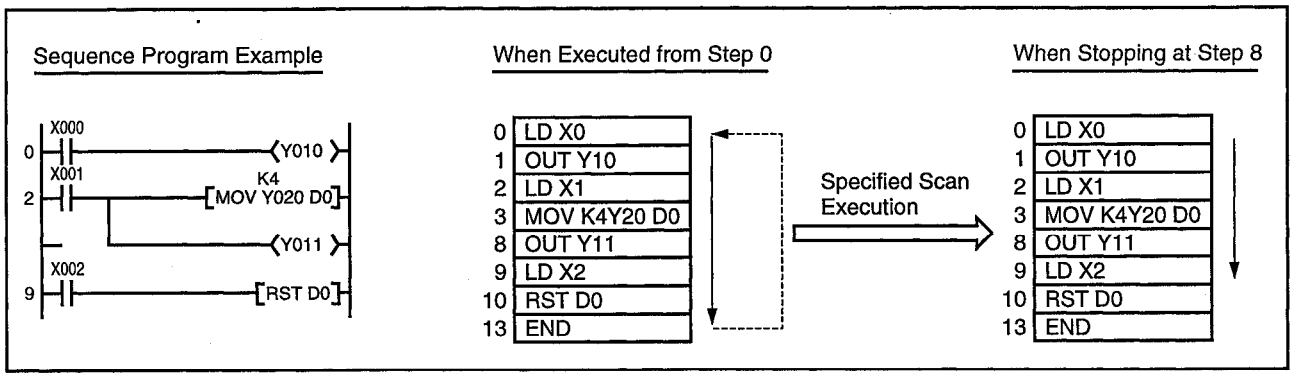
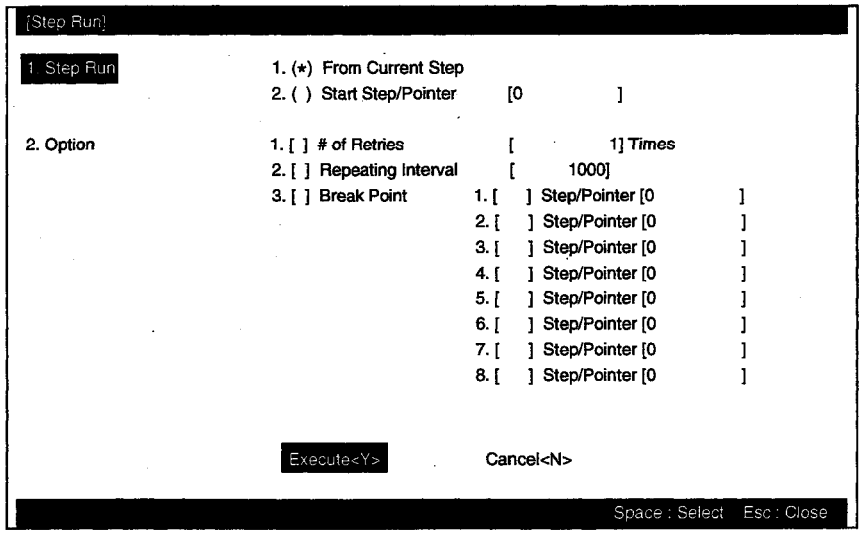


Fig. 8.3 Step-By-Step Execution for the Number of Times to Repeat

Operation Procedure

The step-by-step execution operation is performed in the following manner.
Each operation is performed in the "Monitor" screen in the circuit mode (debug).

(1) Select "B/ Step Run".


[Step Run]
1. Step Run
1. (*) From Current Step
2. ( ) Start Step/Pointer [0]
2. Option
1. [ ] # of Retries [1] Times
2. [ ] Repeating Interval [1000]
3. [ ] Break Point 1. [ ] Step/Pointer [0]
2. [ ] Step/Pointer [0]
3. [ ] Step/Pointer [0]
4. [ ] Step/Pointer [0]
5. [ ] Step/Pointer [0]
6. [ ] Step/Pointer [0]
7. [ ] Step/Pointer [0]
8. [ ] Step/Pointer [0]
Execute<Y>
Cancel<N>
Space : Select Esc : Close

8.7.2 Partial Execution

The program is executed from the presently stopped step or the starting step to the specified step (break point), then stops.

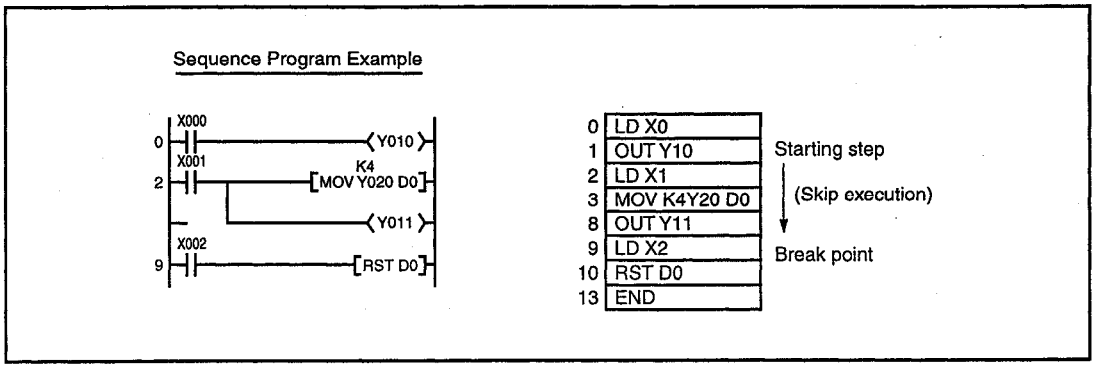


Fig. 8.4 Partial Execution

Operation Procedure

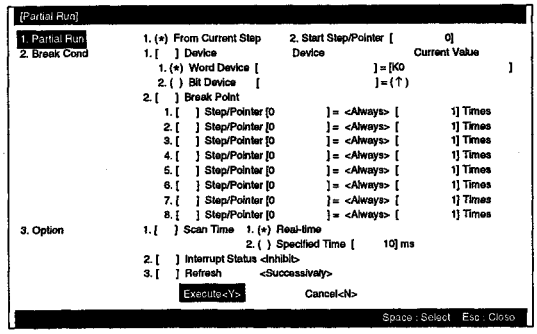
Partial execution operation is performed in the following manner.

Each operation is performed in the "Monitor" screen in the circuit mode (debug).

(1) The execution start step, break condition, and execution operation are specified in the GPP function.

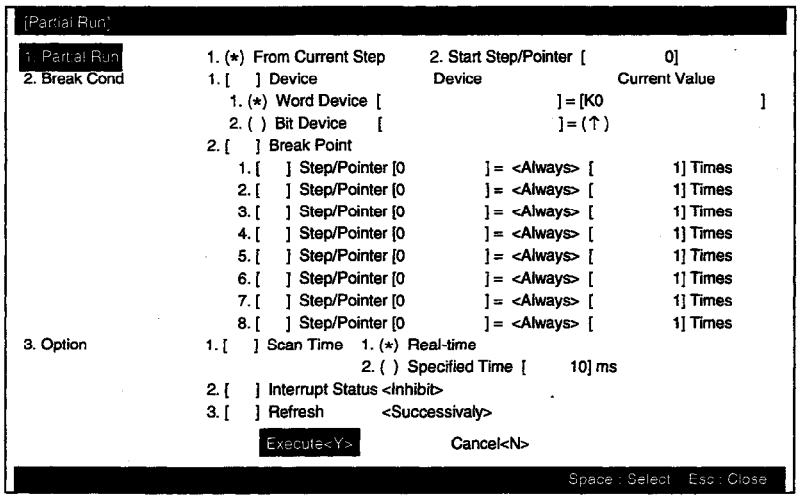
(a) Setting the Execution Starting Step

The step to start the partial execution is specified in "1. Partial Run" in the "Partial Run" screen.



(b) Setting the Break Condition

The break point and device status are set in "2. Break Cond". in the "Partial Run" screen.



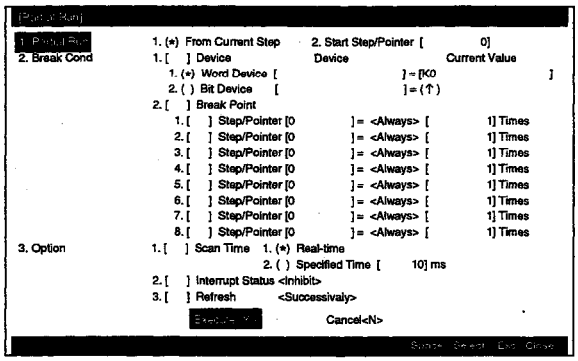
The device that can be set are as follows:

1) Bit Device : X, FX, Y, FY, M, L , F, SM, V, B, and SB.

2) Word Device : T (contact, current value), ST (contact, current value), C (contact, current value), D, SD, FD, W, SW, R, ZR, U□\G, J□\X, J□\Y, J□\B, J□\SB, J□\W, J□\SW, and BL□\S.

(c) Setting the Execution Operation

The scan time, interrupt status, and refresh are specified in "3. Option" in the "Partial Run" screen.



* Multiple items can be set.

Each item in the setting is described below:

Item	Operation (Setting) Description
Scan Time	Specifies whether to execute the scan time in real time or in specified time during partial execution. (Default is specified time 10 ms.)
Interrupt Status	Specifies whether to enable/disable the interrupt status during execution. (Default is disable.)
Refresh	Specifies whether to perform I/O refresh every time the program execution is stopped when the condition is met, or perform refresh only at END. (Default is every time.)

8.7.3 Step-By-Step Execution

The specified step is skipped during execution in the step-by-step execution or partial execution.

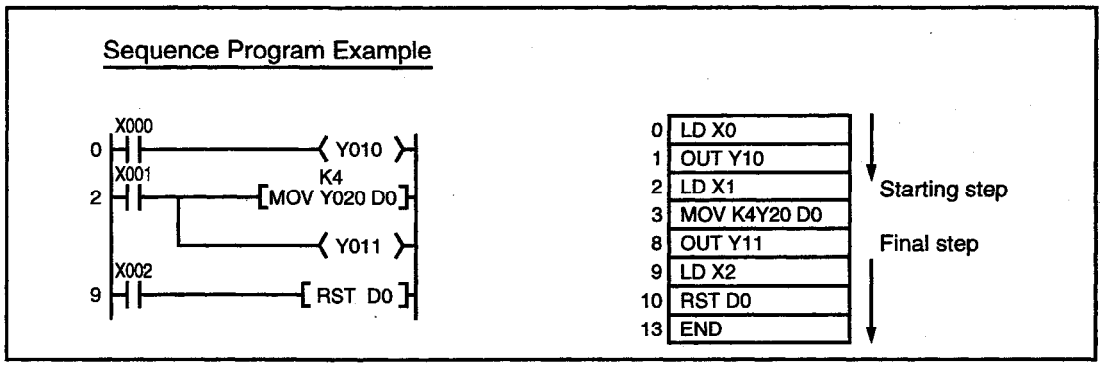


Fig. 8.5 Skip Execution

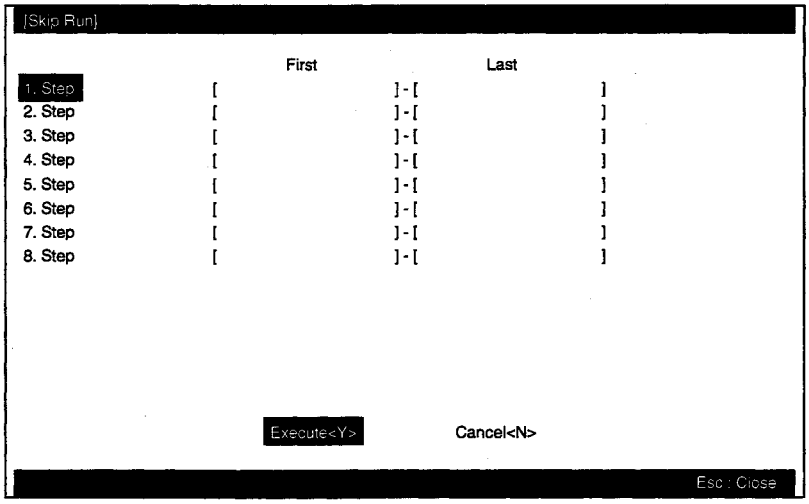
Operation Procedure

Skip execution is performed in the following manner.

Each operation is performed in the "Monitor" screen in the circuit mode (debug).

(1) The program range to skip with the GPP function is set.

The step number to skip is specified in the "D/ Skip Run" screen.



8.8 Program Trace Function

This is a function to sample execution status in the program.

POINT

The memory card is required when executing the program trace function.

Application

The execution status of the specified step in the program can be checked during debug. The debug time can be minimized using this function.

Function Description

(1) Function

(a) The program trace samples the execution status of the specified step in the specified program. This is stored in the program trace file in the memory card.

(b) The devices that can be traced are as follows:

1) Bit Device : X, FX, DX, Y, FY, DY, M, L , F, SM, V, B, SB, T (contact), T (coil), ST (contact), ST (coil), C (contact), C (coil), J□\X, J□\Y, J□\B, J□\SB, and BL□\S ... Maximum 50

2) Word Device : T (current value), ST (current value), C (current value), D, SD, FD, W, SW, R, Z, ZR, U□\G, J□\W, and J□\SW ..... Maximum 50

(c) The program trace file stores the trace condition data and trace execution data necessary to perform the program trace. When trace is started with the GPP function peripheral device, the trace is performed for the number of times set.

(d) The trace number, program name, step number, and device status are displayed in the trace result.

[Trace Results Display]

Times	Program	Step	BranchIns	Time(ms)	D0	M0	M1
-5	MAIN	0	S 0 Step	1891.2	31339		
-4	MAIN	9	END (FEND)	1894.5	31340		
-3	MAIN	0	(Stat)	1894.9	31340		
-2	MAIN	0	S 0 Step	1900.4	31340		
-1	MAIN	9	END (FEND)	1900.6	31341		
0	MAIN	0	(Stat)	1901.1	31341		
1	MAIN	0	S 0 Step	1906.7	31341		
2	MAIN	9	END (FEND)	1906.9	31342		
3	MAIN	0	(Stat)	1707.5	31342		
4	MAIN	0	S 0 Step	1912.7	31342		
5	MAIN	9	END (FEND)	1912.9	31343		
6	MAIN	0	(Stat)	1913.4	31343		
7	MAIN	0	S 0 Step	1918.6	31343		
8	MAIN	9	END (FEND)	1918.8	31344		
9	MAIN	0	(Stat)	1919.3	31344		

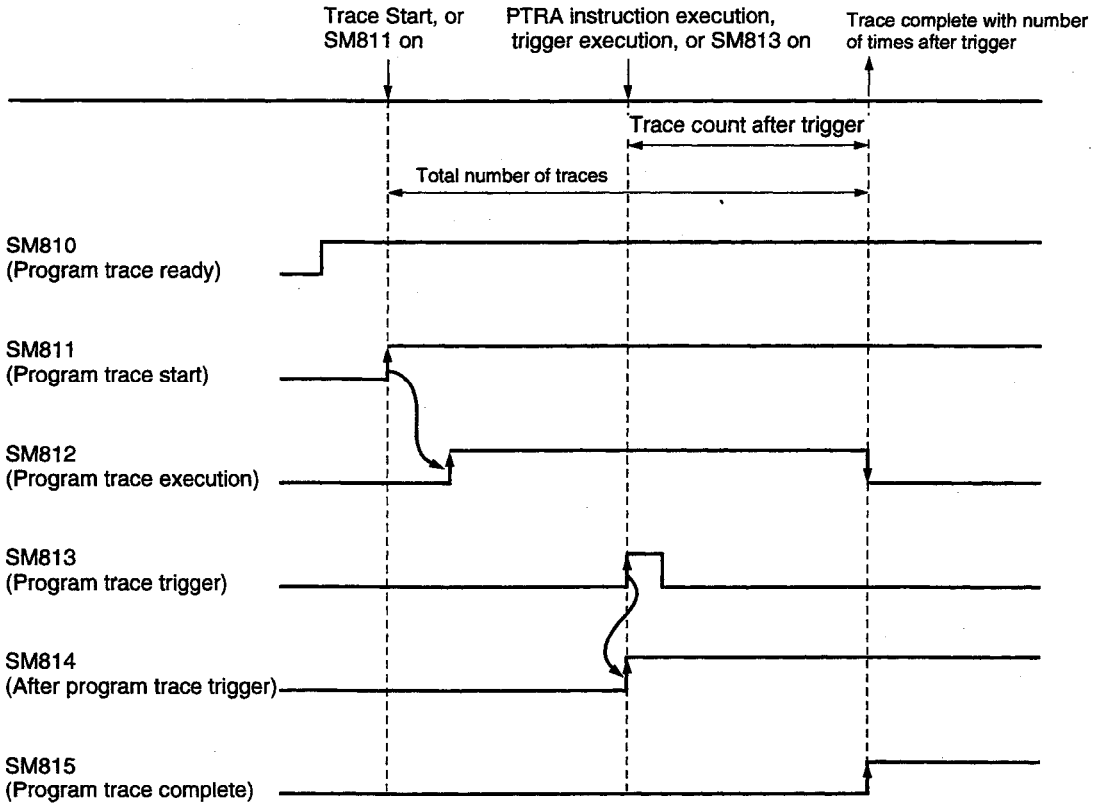
PgUp : Prev PgDn : Next Esc : Close

(2) Basic Operation

The basic operation of the program trace is as follows.

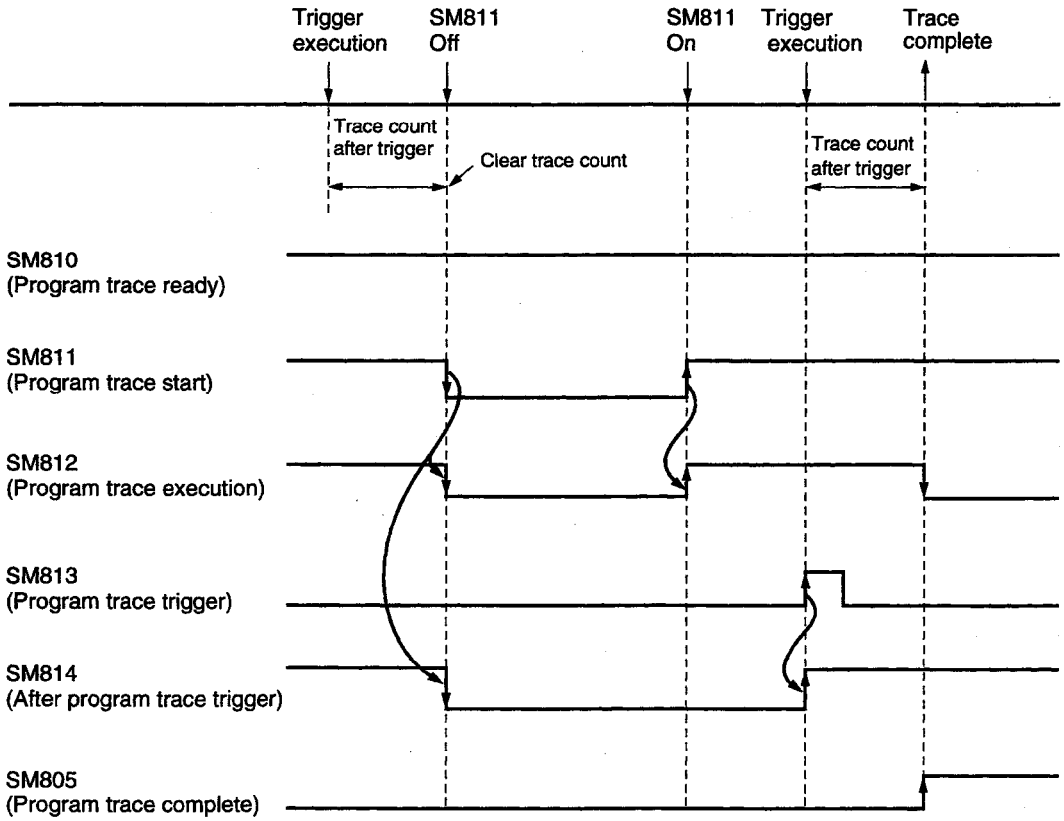
The execution status of the program trace function can be checked in the special relay SM810 to SM815 and SM828.

- When not interrupting the trace


Trace Start, or SM811 on
PTRA instruction execution, trigger execution, or SM813 on
Trace complete with number of times after trigger
Trace count after trigger
Total number of traces
SM810 (Program trace ready)
SM811 (Program trace start)
SM812 (Program trace execution)
SM813 (Program trace trigger)
SM814 (After program trace trigger)
SM815 (Program trace complete)

* SM810 automatically turns on when the program trace is ready.

- When interrupting the trace



* When trace is interrupted from the GPP function peripheral device, the SM810 is turned off also.

The operation when an error occurs is as follows.
When an error occurs during program trace turn, the SM828 (program trace error) is turned on and turn SM811 (program trace start) off.
To turn off SM828, turn on SM811 or execute PTRA instruction.

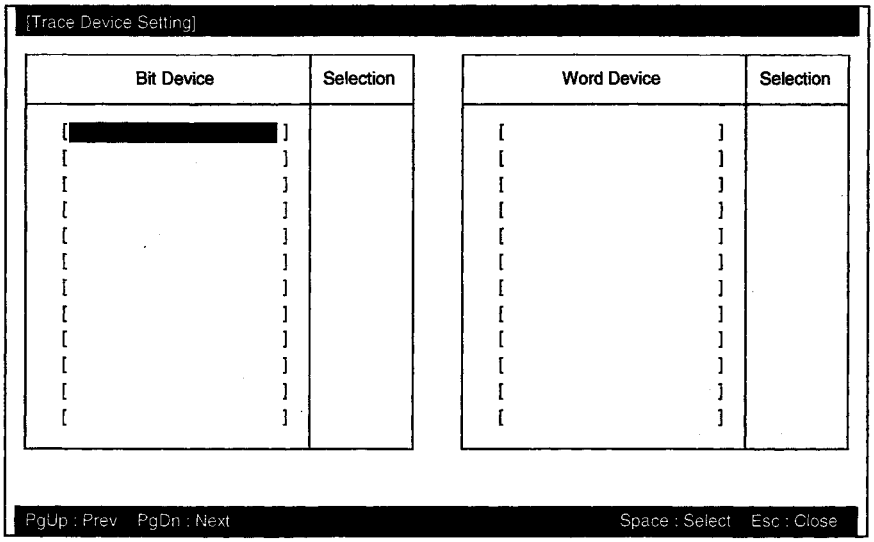
Operation Procedure

The program trace is performed in the following manner.
Each operation is performed in the "Program Trace" screen in the online mode trace menu.
Set the CPU at STEP-RUN at this time. (Refer to Section 8.7.)

(1) Set the trace device and trace condition with the GPP function.

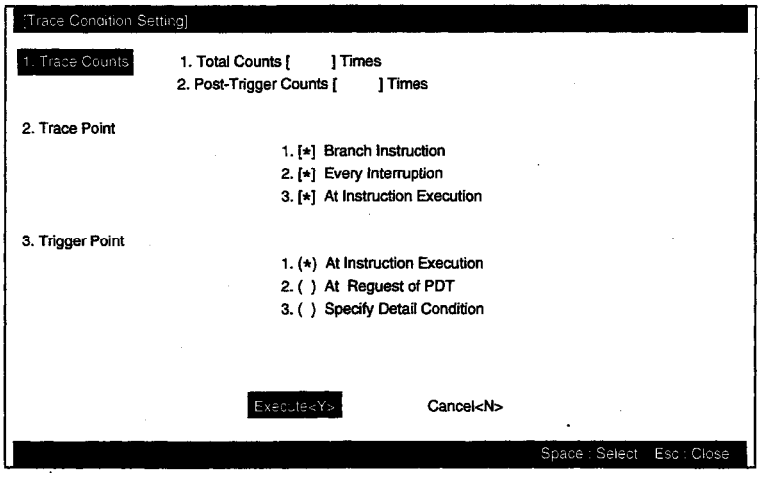
(a) Trace Device Setting

Set the device in "Trace Device Setting" in the "Program Trace" screen.



(b) Trace Condition Setting

Set the trace condition in the "Trace Condition Setting" in the "Program Trace" screen.


[Trace Condition Setting]
1. Trace Counts
1. Total Counts [ ] Times
2. Post-Trigger Counts [ ] Times
2. Trace Point
1. [*] Branch Instruction
2. [*] Every Interruption
3. [*] At Instruction Execution
3. Trigger Point
1. (*) At Instruction Execution
2. ( ) At Reguest of PDT
3. ( ) Specify Detail Condition
Execute<Y>
Cancel<N>
Space : Select Esc : Close

The following describes the screen.

"1. Trace Counts", "2. Trace Point", "3. Trigger Point" can be set in the trace condition setting.

1) Trace Counts

The total count sets the number of executions for the sampling trace from trace start to end.

Count after trigger sets the number of executions for the sampling trace from trigger execution to end of trace.

The setting range for each count is shown below:

Count after trigger ≤ Total count ≤ 8192

2) Trace Point Setting

Set the point to perform trace. Multiple items can be selected from the following:

a) Branch Instruction : Executes for instructions such as CALL and JMP.

b) Every Interruption : Executes for every interrupt program.

c) At Instruction Execution : Executes for every PTRAEXE instruction.

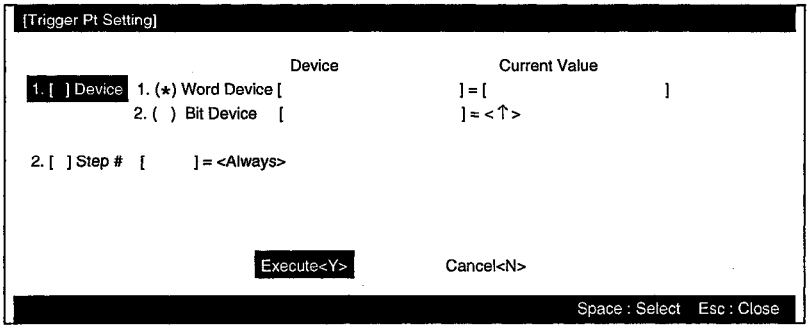
3) Trigger Point Setting

This sets the point to execute the trigger. Select one from the following:

a) At Instruction Execution : The trigger is when the PTRA instruction is executed.

b) At Request of PDT : The trigger is when a trigger operation is performed from the GPP function peripheral device.

c) Specify Detail Condition : Sets the device and step no. The setting example is shown below. The setting method and trigger execution timing is the same as setting the monitor condition in Section 8.2, "Monitor Functions".



The devices that can be set in the detailed condition is as follows:

- Bit Device : X, FX, Y, FY, M, L , F, SM, V, B, SB, T (contact), ST (contact), C (contact), J□\X, J□\Y, J□\B, J□\SB, and BL□\S.
- Word Device : T (current value), ST (current value), C (current value), D, SD, FD, W, SW, R, Z, ZR, U□\G, J□\W, and J□\SW.

The following qualification can be set for the above devices:

- Bit device number of digits specification
- Word device bit number specification

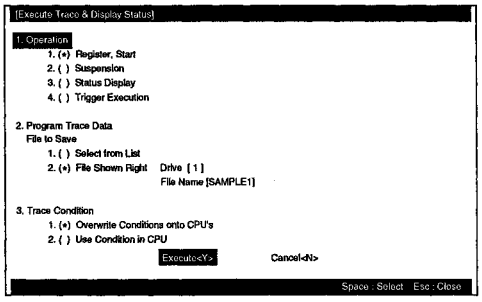
POINT

The time, program name, step and branch factor when the trace was executed are added automatically.

(2) The trace device and trace condition created is written into the memory card.

(a) Set the trace file and storage destination.

The file name and drive number are set in "1. ( ) Execute Trace & Display Status" in the "Program Trace" screen.


[Execute Trace & Display Status]
1. Operation
1. (•) Register, Start
2. ( ) Suspension
3. ( ) Status Display
4. ( ) Trigger Execution
2. Program Trace Data
File to Save
1. ( ) Select from List
2. (•) File Shown Right Drive [1]
File Name [SAMPLE1]
3. Trace Condition
1. (•) Overwrite Conditions onto CPU's
2. ( ) Use Condition in CPU
Execute<Y>
Cancel<N>
Space : Select Esc : Close

(b) Write the trace file into the memory card.

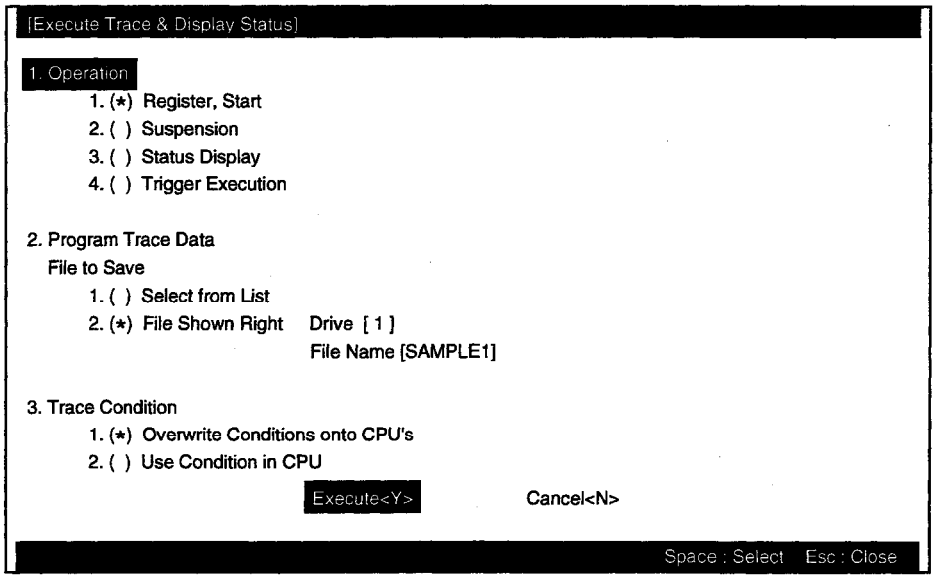
The trace file is written in the memory card from "9. ( ) Write to PC (Condition)" in the "Program Trace" screen.

The file name is written in the memory card, so multiple trace files can be stored.

(3) Program trace is executed.

Program trace is executed at "1. ( ) Execute Trace & Display Status" in the "Program Trace" screen.

The setting example of "1. ( ) Execute Trace & Display" is shown below:


[Execute Trace & Display Status]
1. Operation
1. (*) Register, Start
2. ( ) Suspension
3. ( ) Status Display
4. ( ) Trigger Execution
2. Program Trace Data
File to Save
1. ( ) Select from List
2. (*) File Shown Right
Drive [1]
File Name [SAMPLE1]
3. Trace Condition
1. (*) Overwrite Conditions onto CPU's
2. ( ) Use Condition in CPU
Execute<Y>
Cancel<N>
Space : Select Esc : Close

The following describes the screen.

"1. Operation", "2. Program Trace Data", and "3. Trace Condition" can be set in the execute trace & display status.

(a) Operation

Select one from the following:

1) Register, Start: : Starts the trace. Starts the trace count.
2) Suspension : Interrupts the trace. The trace count and count after trace are cleared. (When restarting trace, select "Trace Start".)
3) Status Display : The trace status are displayed as messages all in one screen.
4) Trigger Execution : Starts the count after trigger. The trace is complete once the count reaches the specified count after trigger.

(b) Program Trace Data

Select one from the following:

1) Select from List : Select from the program trace file in the memory card.
2) File Shown Right : Set the drive number and program trace file name.

(c) Trace Condition

Select one from the following:

1) Overwrite Conditions onto CPU's : Overwrites the trace condition in the existing race file.
2) Use Condition in CPU : Executes with the condition in the trace file specified in "2. Program Trace Data".

(4) Read the trace result from the CPU module and displays the result.
   (a) The trace result is read from the CPU module with "A. ( ) Read from PC (Results)" in the "Program Trace" screen.
   (b) The trace result read from "4. ( ) Trace Results Display" in the "Program Trace" screen and is displayed.

POINT

When program trace is executed once, the trace is not executed second time. When re-executing, execute the PTRAR instructions, and reset the program trace.

Precaution

(1) The program trace can only be executed at STEP-RUN.
(2) Set the program trace file in the RAM of the memory card.
(3) The program trace can be executed from another station on the network or serial communication module. However, the trace cannot be executed from multiple areas at once. The trace can only be executed from one area with Q2ASCPU.
(4) This is performed by connecting the Q2ASCPU and GPP function peripheral device.

8.9 Simulation Function

Application

This is a function to simulate the program in step-by-step execution or partial execution by separating the input module, output module, and special function module from the CPU module. By using this function, the program can be debugged without affecting each module.

Function Description

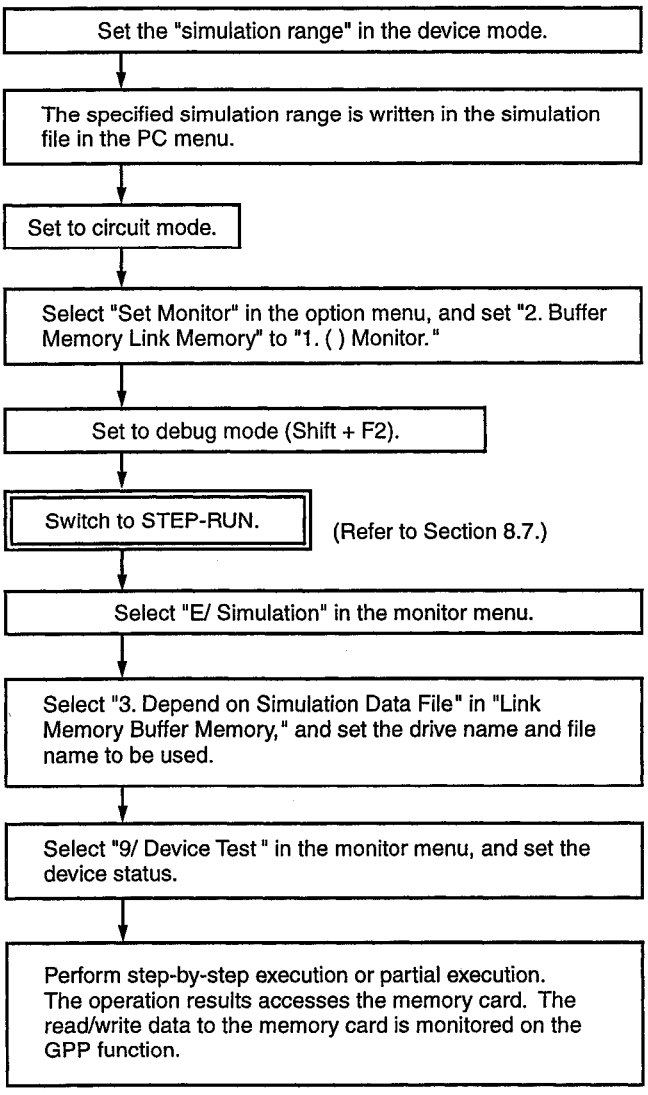
(1) Set the input refresh and output refresh not to be performed for the input or output modules during program execution. By doing so, the input from an external source and output to the external source can be separated.

(2) Set "ignore"/"according to the simulation data file" for the special function module buffer memory to separate this from the special function module operations.

Operation Procedure

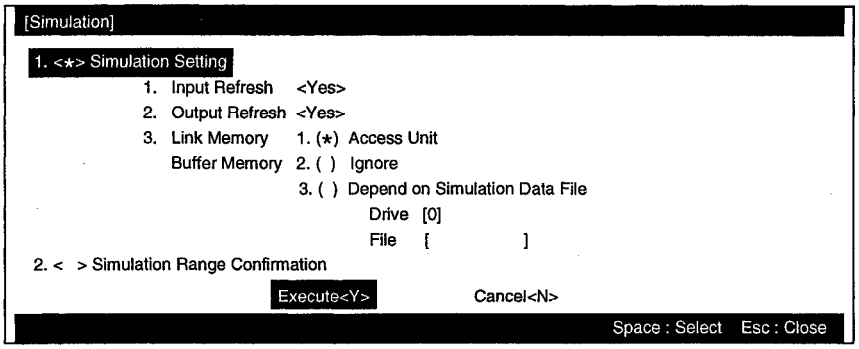
The simulation procedure is shown below:

☐ indicates the GPP function operation and, ▭ indicates the CPU module operation.


Set the "simulation range" in the device mode.
The specified simulation range is written in the simulation file in the PC menu.
Set to circuit mode.
Select "Set Monitor" in the option menu, and set "2. Buffer Memory Link Memory" to "1. ( ) Monitor."
Set to debug mode (Shift + F2).
Switch to STEP-RUN.
(Refer to Section 8.7.)
Select "E/ Simulation" in the monitor menu.
Select "3. Depend on Simulation Data File" in "Link Memory Buffer Memory," and set the drive name and file name to be used.
Select "9/ Device Test " in the monitor menu, and set the device status.
Perform step-by-step execution or partial execution. The operation results accesses the memory card. The read/write data to the memory card is monitored on the GPP function.
(Refer to Section 8.7.)

(*) Described in this Section.

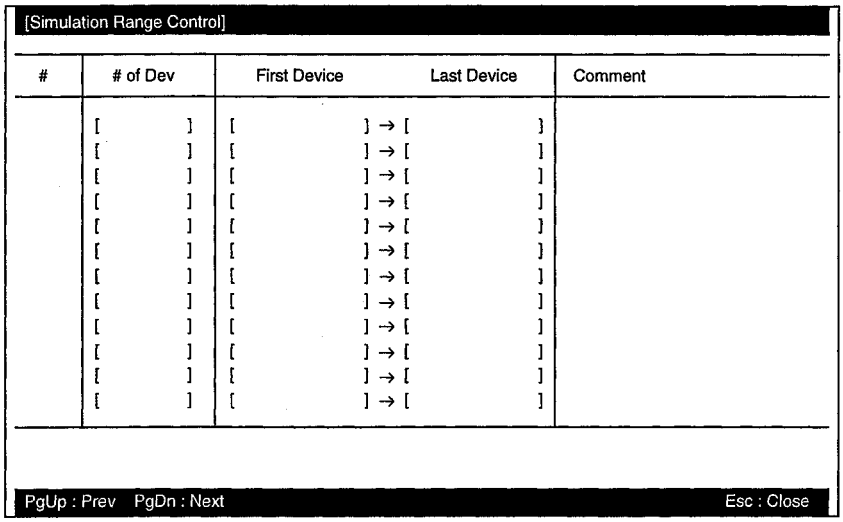
(1) The setting is performed in the following simulation setting screen:



- The details of each setting item is shown below:

Setting Item	Select range	Description
Input Refresh	Yes/No	Selects whether to read in the input from an external source in the CPU module.
Output Refresh	Yes/No	Selects whether to output the CPU module internal operation results to an external source.
Link Memory/ Buffer Memory	Access/ignore module. Follows the simulation data file.	Selects the access method to each module.

When "Depend on Simulation Data File" is selected in the link memory/buffer memory, the access range can be check for each module when performing a simulation range setting check.



Precaution

(1) The simulation can be executed in the STEP-RUN mode only.

(2) A memory card is necessary when performing the link memory/buffer memory simulation with the simulation data file.

(3) The simulation can be executed from another station on the network or serial communication module. However, the simulation cannot be executed from multiple areas at once. The simulation can only be executed from one area with Q2ASCPU.

(4) The simulation is performed by connecting the Q2ASCPU and GPP function peripheral device.

(5) Take Note of the following when executing the simulation:
- When direct input (DX) and direct output (DY) are used to handle the I/O directly, the actual I/O is not accessed and accesses the device memory.
- All special function module instructions are not processed.
- When "SP. UNIT ERROR" occurs, FFFFH is displayed in the shared data module no. column.
- When the buffer memory access method is set to "ignore", the command access or monitor results will be FFFFH.

8.10 Debug Function with Multiple Users

This is a function to execute debug from multiple GPP function peripheral devices at the same time.

Application

This is used when debugging from different files from multiple GPP function peripheral device is performed at once.

Function Description

The debug function combination for multiple users are as follows:

Debug Function from Other Station / Debug Function from Host	Monitor	Write during RUN	Execution time measurement	Sampling trace/ program trace	Status latch	Step-by-step operation	Simulation
Monitor	○	×	○	○	○	○	○
Write during RUN	×	○	×	×	×	×	×
Execution time measurement	○	×	×	○	○	○	○
Sampling trace	○	×	○	×	○	○	○
Program trace	○	×	○	×	○	○	○
Status latch	○	×	○	○	×	○	○
Step-by-step operation	○	×	○	○	○	×	○
Simulation	○	×	○	○	○	○	×

○: Can be executed at the same time. (However, the detailed condition can only be set from one GPP function peripheral device. In this case, the detailed condition setting cannot be performed from another GPP function peripheral device.)

×: Can only be executed from one GPP function peripheral device.

8.10.1 Multiple-User Monitoring Function

The Q2ASCPU allows multiple users to monitor at the same time. By setting a station monitor file to the system area of the built-in RAM, high-speed monitoring can be performed from the other station. (Setting of station monitor file is not necessary.)

Operation Procedure

The multiple-user monitoring operation is performed in the following procedure:

(1) Select "5. ( ) Format <with Option>" in "B/PC Memory Batch Processing" in the online mode "2/ PC" and set the station monitor file.

The setting example is shown below:

A maximum of 15 k steps can be set in 1 k step units as a system area. Only 1 k step can correspond to one station monitor file.

Therefore, a maximum of 15 station monitor files can be set.

The program file area of the built-in RAM is in the same area as the station monitor file, so the program file area is reduced when station monitor files are set.

Precaution

(1) The detailed condition setting of the monitor can only be set from one area.

(2) Monitoring can be performed even if a station monitor file is not set, but high-speed monitoring cannot be performed.

(3) When it is desired to perform simultaneous monitoring from multiple users at high speed, perform this operation before writing a parameter file, program file, etc. to the built-in RAM. If this operation is performed after the files are written to the built-in RAM, all files will be erased.

(4) Up to 16 locations per CPU module can be accessed simultaneously.

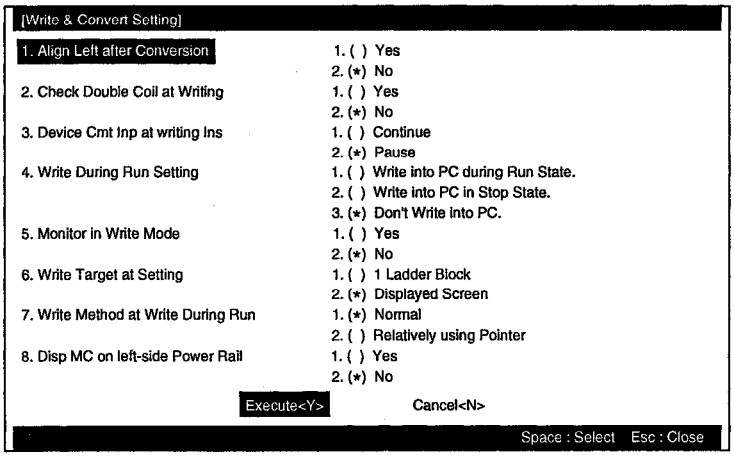
8.10.2 Multiple-User RUN Write Function

The Q2ASCPU allows multiple users to write to one file or different files during RUN.

Operation Procedure

The multiple-user RUN write operation is performed in the following manner:

(1) Set "4. Write During Run Setting" and "7. Write Method at Write During Run" are set in the circuit mode "8/ Option" menu.
The setting example is shown below.

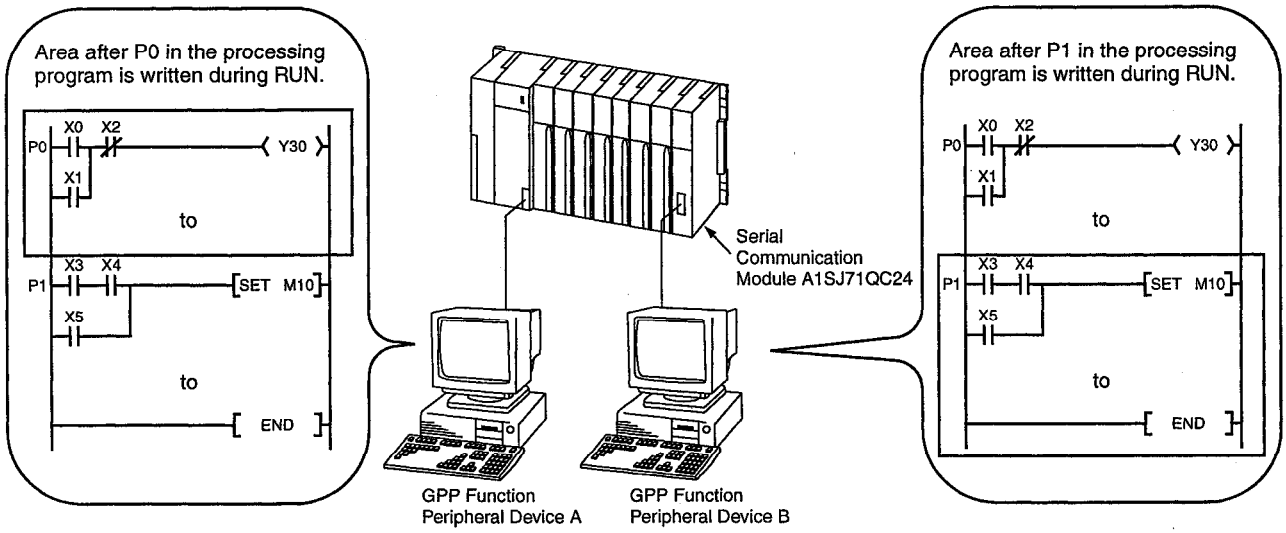

[Write & Convert Setting]
1. Align Left after Conversion
1. ( ) Yes
2. (*) No
2. Check Double Coil at Writing
1. ( ) Yes
2. (*) No
3. Device Cmt Inp at writing Ins
1. ( ) Continue
2. (*) Pause
4. Write During Run Setting
1. ( ) Write into PC during Run State.
2. ( ) Write into PC in Stop State.
3. (*) Don't Write into PC.
5. Monitor in Write Mode
1. ( ) Yes
2. (*) No
6. Write Target at Setting
1. ( ) 1 Ladder Block
2. (*) Displayed Screen
7. Write Method at Write During Run
1. (*) Normal
2. ( ) Relatively using Pointer
8. Disp MC on left-side Power Rail
1. ( ) Yes
2. (*) No
Execute<Y>
Cancel<N>
Space : Select Esc : Close

(a) Set "1. ( ) Write into PC During Run State" in "4. Set Write/ Convert".

(b) Select "1. ( ) Normal" or "2. ( ) Relatively Using Pointer" in "Write Method at Write During Run".

When multiple users are writing at the same time during RUN to one file, set a pointer to write during RUN beforehand, and select "2. ( ) Relatively Using Pointer".

The following is an example of GPP function peripheral device A writing during RUN from P0 and GPP function peripheral device B writing during RUN from P1. The program area surrounded with ☐ is the area to be written during RUN.



Precaution

Refer to Section 8.3.

9 Maintenance Functions

9.1 Function List

The following is the list of maintenance functions:

Item		Description	Reference
Watchdog timer		Function to supervise the watchdog error due to CPU module hardware and program errors.	Section 9.2
Self-diagnosis function		Function to diagnose errors in Q2ASCPU.	Section 9.3
Failure history		Function to store the diagnosis results in the failure history.	Section 9.4
System protect		Function to set enable/disable of the read/write for the Q2ASCPU file.	Section 9.5
Password registration		Function to prohibit operations of the CPU module memory from GPP function.	Section 9.6
System display		Function to monitor the system configuration by connecting the GPP function peripheral device.	Section 9.7
LED display		Function to display the CPU module operation status with the LED located on the front of the CPU module.	Section 9.8
	LED display	Displays whether the CPU module operation is normal or has an error.	Section 9.8.1
	Priority setting	Sets the LED display priority according to the error definition.	Section 9.8.2

Refer to "GX Developer OPERATING MANUAL" or "SW□IVD-GPPQ OPERATING MANUAL (Online)" for details of the GPP function operation method.

POINT

An I/O module that is online cannot be replaced.

9.2 Watchdog Timer

(1) Watchdog Timer (WDT)

The watchdog timer is an internal sequence timer to detect PC hardware and/or sequence program error.

The default value is set at 200 ms.

Remarks

The watchdog timer set time can be changed in "WDT Setting" in the GPP function parameter mode PC RAS setting.

The setting range is 10 to 2000 ms (10ms units).

(2) Watchdog Timer Reset

The Q2ASCPU resets the watchdog timer (during END processing execution).

When the Q2ASCPU is operating correctly and the END instruction is executed within the set value of the watchdog timer, the watchdog timer does not time out.

The watchdog timer times out when there is a Q2ASCPU hardware fault or the sequence program scan time is too long to execute the END instruction within the set value of the watchdog timer.

Remarks

The scan time is the time when the PC starts executing the sequence program by performing the calculation from step 0 and then completes the program and executes step 0 again.

The scan time is not the same at every scan, and differs, depending on whether the commands used are executed or not executed. (Refer to Section 12.1.)

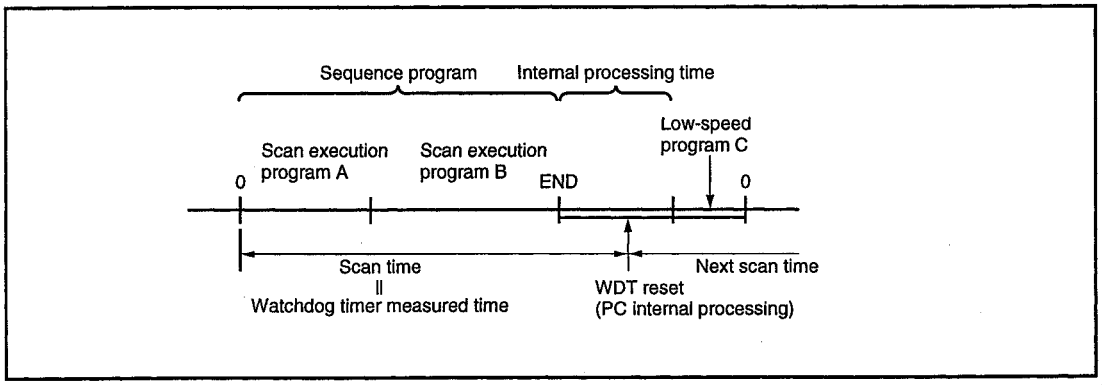


Fig. 9.1 Watchdog Timer Reset

(3) Processing when the Watchdog Timer Times Out

When the scan time exceeds the set watchdog timer value, watchdog time error results and the PC behaves in the following manner:

(a) The PC turns off all output.

(b) The RUN LED located on the front of the CPU module turns off and flashes the ERROR LED.

(c) SM1 turns on and the error code is stored in SD0.

Remarks

The watchdog timer is reset with the WDT instruction in the sequence program. However, the scan time value is not reset, and measures to the END instruction.

POINT

The watchdog timer measurement time can yield an error between 0 to 10 ms.

9.3 Self-Diagnosis Function

The self-diagnosis function is a function performed by the Q2ASCPU itself to diagnose whether there is an error in the Q2ASCPU.

(1) The self-diagnosis function's objective is to prevent the PC erroneous operation and as preventive maintenance.
The self-diagnosis processing detects and displays the error when an error occurs when the PC power is turned on or during PC RUN mode. It also stops PC calculations.

(2) The Q2ASCPU stores the error code of the error in the special register SD0, and turns on the ERROR LED and displays a message.
When multiple errors occur, the latest error code is stored in the SD0.

(3) 16 latest errors can be stored even when the PLC is powered off using the battery backup. (Refer to Section 9.4.)
The failure history can be checked in the GPP function PC diagnosis mode.

(4) When an error is detected from the self-diagnosis, there are two types of modes that the CPU module operation can change to.
- PC calculation stop mode
  Stops the calculation at the point when the error is detected, and turns off all output (Y).
- PC calculation continue mode
  When an error is detected, the program area where the error occurred are skipped and executes the rest of the program.

The following errors can set the calculation "continues/stops" in the parameter mode PC RAS setting.
(a) Calculation error (including the SFC program)
(b) Extension instruction error
(c) Fuse short
(d) I/O module verification error
(e) Special function module error
(f) File access error
(g) Memory card operation error
(All parameter defaults are set at "Stops".)

For example, when the I/O module verification error is set to "continues", the calculations are continued in the I/O address before the error occurred.
When an error is detected, error occurrence is stored in the special relay (SM0 and SM1) and error details are stored in the special register (SD0). Interlock the devices and PC using the special relay and special register.

(5) The error checking can be set to "yes/no" in the following error checking in the parameter mode PC RAS setting:
(a) Battery check
(b) Fuse short check
(c) I/O module verification
(All parameter defaults are set at "Yes".)
When error checking is set to "No", error detection is not performed so the processing time to the END instruction can be minimized.

Self-Diagnosis List

Diagnosis description		Diagnostic timing	CPU module status	LED status	
				RUN	ERROR
Hardware failure	CPU module error	Always	Stop	Off	Flashing
	END instruction not executed	When the END instruction is executed	Stop	Off	Flashing
	RAM check	When the power is turned on/when reset	Stop	Off	Flashing
	Calculation circuit check	When the power is turned on/when reset	Stop	Off	Flashing
	Fuse short (default... stop) *1	When the END instruction is executed (default... Yes) *2	Stop/Continue	Off/On	Flashing/On
	I/O interrupt error	When an interrupt occurs	Stop	Off	Flashing
	Special function module error	• When the power is turned on/when reset • When the FROM/TO instruction is executed	Stop	Off	Flashing
	Control bus error	• When the power is turned on/when reset • When the FROM/TO instruction is executed	Stop	Off	Flashing
	Momentary stop occurred	Always	Continue	On	Off
	Battery low	Always (Default... Yes)*3	Continue	On	Off
Handling error	I/O module verification (Default... Stop) *1	When the END instruction is executed (Default... Yes)*2	Stop/Continue	Off/On	Flashing/On
	Special function module allocation error	• When the power is turned on/when reset • When switched from STOP to RUN	Stop	Off	Flashing
	Special function module error (Default... Stop)*1	When the FROM/TO instruction is executed.	Stop/Continue	Off/On	Flashing/On
	No parameter	When the power is turned on/when reset	Stop	Off	Flashing
	Boot error	When the power is turned on/when reset	Stop	Off	Flashing
	Memory card operation error (Default... Stop) *1	When the memory card is installed/ removed	Stop/Continue	Off/On	Flashing/On
	File setting error	When the power is turned on/when reset	Stop	Off	Flashing
	File access error (Default.. Stop) *1	When an instruction is executed	Stop/Continue	Off/On	Flashing/On
	Instruction execution not possible	When the power is turned on/when reset	Stop	Off	Flashing
Parameter error	Parameter setting check	• When the power is turned on/when reset • When switched from STOP to RUN	Stop	Off	Flashing
	Link parameter error	• When the power is turned on/when reset • When switched from STOP to RUN	Stop	On	On
	SFC parameter error	When switched from STOP to RUN	Stop	Off	Flashing

*1: Can be changed to "Continue" in the GPP function parameter setting.
*2: Can be set to "No" in the GPP function parameter setting. Also, checking is not performed when SM251 is on.
*3: Can be set to "No" in the GPP function parameter setting.

Diagnosis description (Continued from the preceding page)

Diagnosis description		Diagnostic timing	CPU module status	LED status	
				RUN	ERROR
Program error	Instruction code check	• When the power is turned on/when reset • When switched from STOP to RUN	Stop	Off	Flashing
	No END instruction	• When the power is turned on/when reset • When switched from STOP to RUN	Stop	Off	Flashing
	Pointer setting error	• When the power is turned on/when reset • When switched from STOP to RUN	Stop	Off	Flashing
	Pointer setting error	• When the power is turned on/when reset • When switched from STOP to RUN	Stop	Off	Flashing
	Operation check error (Default... Stop) *1	When an instruction is executed	Stop/Continue	Off/On	Flashing/On
	FOR to NEXT instruction structure error	When an instruction is executed	Stop	Off	Flashing
	CALL to RET instruction structure error	When an instruction is executed	Stop	Off	Flashing
	Interrupt program error	When an instruction is executed	Stop	Off	Flashing
	Instruction execution not possible	When an instruction is executed	Stop	Off	Flashing
	Extension instruction error (Default... Stop) *1	When an instruction is executed	Stop/Continue	Off/On	Flashing/On
	SFC program structure error	When switched from STOP to RUN	Stop	Off	Flashing
	SFC block structure error	When switched from STOP to RUN	Stop	Off	Flashing
	SFC step structure error	When switched from STOP to RUN	Stop	Off	Flashing
	SFC syntax error	When switched from STOP to RUN	Stop	Off	Flashing
	SFC operation check error (Default... Stop) *1	When an instruction is executed	Stop/Continue	Off/On	Flashing/On
	SFC program execution error	When switched from STOP to RUN	Continue	On	On
	SFC block execution error	When an instruction is executed	Stop	Off	Flashing
	SFC step execution error	When an instruction is executed	Stop	Off	Flashing
CPU error	Watchdog error supervision	Always	Stop	Off	Flashing
	Program time exceeded	Always	Continue	On	On
Annunciator check		When an instruction is executed	Continue	On	Off
CHK Instruction check		When an instruction is executed	Continue	On	Off

*1: Can be changed to "continues" in the GPP function parameter setting.

9.3.1 Interrupt due to Error Occurrence

The Q2ASCPU can execute the interrupt pointer interrupt program set when an error occurs. An error that can set the calculations to "continues/stops" in the GPP funciton parameter mode PC RAS setting executes the calculation for the error set as "continues". The error set to "stops" executes the stop error interrupt program (I32).
The errors are shown below:

Interrupt pointer	Corresponding error message
132	Stop all errors
133	Vacancy
134	UNIT VERIFY ERR. FUSE BREAK OFF SP. UNIT ERROR
135	OPERATION ERROR SFCP OPE. ERROR SFCP EXE. ERROR
136	ICM. OPE. ERROR FILE OPE. ERROR
137	EXTEND INS. ERR.
138	PRG. TIME OVER
139	CHK instruction Annunciator detect
140 to 147	Vacancy

When the error occurs and the system can continue the drive mode. Or it is an error where "continues/stops" can be selected, and "continues" is set.

POINT

The interrupt pointers I32 to I39 is at an execution disable mode when the PLC CPU's power is on or CPU module is reset. When using I32 to I39, use the IMASK instruction to enable execution.

Remarks

(1) Refer to the QnACPU Programming Manual (Fundamentals) for details of the interrupt pointers.
(2) Refer to the QCPU (Q Mode)/QnACPU Programming Manual (Common Instructions) for the IMASK instruction.

9.3.2 LED Display when Error Occurs

When an error occurs, the LED located on the front of the CPU module turns on.
Refer to Section 9.8 for the details of the LED display.

9.3.3 Cancel Error

Q2ASCPU error cancel operation can be performed only for error that can continue the CPU module operation.

The error cancel is performed in the following manner:

(1) Resolve the cause of error.

(2) Store the error code of the error to be canceled in the special register SD50.

(3) Switch special relay SM50 from OFF → ON.

(4) The error is canceled.

When the CPU module is recovered from canceling the error, the special relay, special register, and LED affected by the error are set to the state before the error occurred.

When the same error occurs after canceling the error, it is logged again in the failure history.

For the cancellation of the enunciator detected multiple times, only the first detected "F" is canceled.

POINT

When error cancellation is performed by storing the error code of the error to cancel is stored in SD50, the lower 2 digits of the code number is ignored.

[Example]

When 2100 and 2111 occur in the error code and error code 2100 is canceled, error code 2111 is canceled as well.

9.4 Failure History

The Q2ASCPU can store the failure history (results detected from the self-diagnosis function and the time) into the memory.

POINT

The detection time uses the Q2ASCPU internal clock, so make sure to set the correct time when first using the CPU module.

(1) Storage Area

- The latest 16 failures are stored in the CPU module failure history storage memory.
- When storing more than 16, the history can be stored in the memory card file using the GPP fucntion parameter mode PC RAS setting.
- When the history count of the parameter and memory card are different when the following operation is performed, clear the contents of the memory card history file, then the Q2ASCPU transfers the 16 failure data in the CPU module failure history storage memory to the file.

(a) When the failure history is changed in the parameter history file.

(b) When a memory card which has a different history count than the parameter is set

- The storage area in the failure history file is as follows:

Storage area	File in the set memory card
Amount that can be stored	Max. 100 (can be changed)*1

*1: When the number of storage exceed the amount that can be stored, the oldest history is overwritten with the latest history.

POINT

Even if the failure history file set in the parameter does not exist in the memory card, the CPU module does not yield an error.

The CPU module only performs the processing to store the failure to the failure history storage file.

(2) Failure History Clearing Method

The failure history is cleared using the failure history clear in the GPP function PC diagnosis mode PC menu.

Failure history clear erases all the failure history storage memory contents of the CPU module and the failure history file contents of the memory card.

9.5 System Protect

The Q2ACPU has a few protection functions (system protect) for the program changes to processing of general data obtained from a third party other than the designer(access processing from GPP function or serial communication module).
There are the following methods for system protect.

Item to protect	Protect valid file	Protection description	Method	Valid Timing	Remarks
All of CPU module	All files	Prohibits all write/control specification to the CPU module.	Set the CPU module system setting switch1 SW1 on. (Refer to Section 15.2)	Always	Valid for devices also
Memory card unit	All files	Performs drive protect for the memory card, and write protect.	Set write protect switch on the memory card on. (Refer to Section 18.5.)	Always	
Drive unit	Parameter program	The following settings are password registered for the specified drive (e.g. internal RAM): (1) Read/Write display prohibit (2) Write prohibit	Perform password registration. (Refer to Section 9.6.)	Always	
File unit	All files	The attributes for a file is changed to the following: (1) Read/Write display prohibit (2) Write prohibit	Change the attributes for the file in the Password Registration. (Refer to Section 9.6.)	Always	

* The control instruction, read/write display, and write are mentioned above are as follows:

Item	Description
Control specification	CPU module operation specification from remote operation. (Remote RUN, remote STOP, etc.)
Read/Write display	Program read/write operations.
Write	Operation that writes the program and tests.

9.6 Password Registration

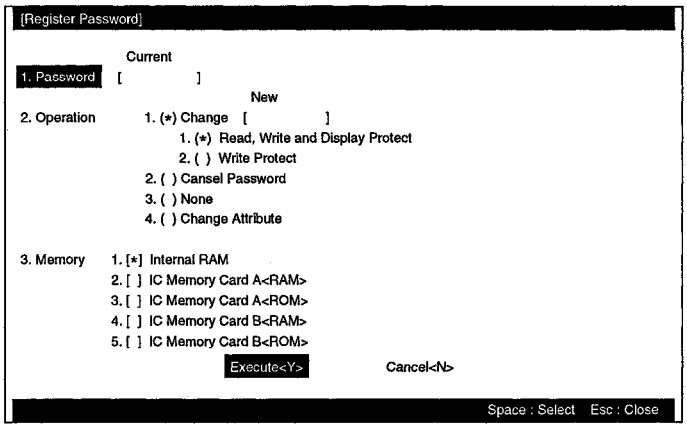
Password is used to prohibit the data read and write of the program and comments in Q2ASCPU from a peripheral device.
The Password Registration is set for the specified memory (internal RAM, memory card) parameter file and program file. There are two description of items to be registered.

- The file name is not displayed and read/write cannot be performed as well.
- Write cannot be performed to the file. (Read only)

If the password is registered, file operations from the peripheral device cannot be performed unless the password registered in the CPU module is input.

(1) Password Registration

The password registration is performed with the password registration in the online mode PC menu of the GPP function.


[Register Password]
Current
1. Password [ ]
New
2. Operation 1. (*) Change [ ]
1. (*) Read, Write and Display Protect
2. ( ) Write Protect
2. ( ) Cansel Password
3. ( ) None
4. ( ) Change Attribute
3. Memory 1. [*] Internal RAM
2. [ ] IC Memory Card A<RAM>
3. [ ] IC Memory Card A<ROM>
4. [ ] IC Memory Card B<RAM>
5. [ ] IC Memory Card B<ROM>
Execute<Y>
Cancel<N>
Space : Select Esc : Close

Each item is described below:

(a) Password.....When the password is already stored in the CPU module, the operation can be performed when the password is entered. If other password is entered, the operation cannot be performed.

(b) Operation.....1) Change : A new password is registered in the CPU module. Or, when the password matches, the password can be changed.

a) Read, Write and Display Protect : The specified file name in the memory is not displayed and read/write cannot be performed.

b) Write Protect : Cannot write to the specified file in the memory. Read only.

2) Cancel Password : When the password matches, the stored password is deleted from the CPU module.

3) None : The present password is stored only in the GPP function, and not registered in the CPU module.

4) Change Attribute: : The read/write/display protect and write protect can be set for each file.
(This operation can be performed even if the password is not registered.)

(c) Memory .....Specifies the memory that stores the password.

POINTS

(1) The file that can register passwords are the parameter files and program files only.

(2) The password stored in the CPU module cannot be read from the CPU module.
If the stored password is forgotten, the CPU module files cannot be manipulated.
Store the registered password in a document for safekeeping.

(3) Password registration uses the memory area of one file.
(4k bytes are used when a password is registered to the built-in RAM.)

9.7 System Display

The following items can be checked in the Q2ASCPU by connecting the GPP function peripheral device.

(1) The following information about the module actually installed on the base:
  (a) Type
  (b) Number of I/O points
  (c) Head X/Y No.

(2) The following information about the parameters set in the module:
  (a) Type
  (b) Number of I/O point
  (c) Model

(3) The following information about the CPU module
  (a) RUN/STOP key switch status
  (b) System setting switch status
  (c) LED on status

Each item can be checked with the module details display in the display menu of the GPP function PC diagnosis mode, or CPU panel.

9.8 LED Display

The Q2ASCPU has an LED to indicate the CPU module operation status on the front of the CPU module. The display details of each LED is described below.

9.8.1 LED Display

(1) The details of the LED display is shown below:

LED name	Display Description
RUN	Indicates the CPU module operation status. On : When operating with the RUN/STOP key switch at "RUN" or "STEP-RUN". Off : When stopped with the key switch at "STOP", "PAUSE", or "STEP-RUN". Or when an error that stops operation is detected. Flashing : The program is written in during STOP, and the RUN/STOP key switch is set from "STOP" to "RUN". To turn on the LED again, set the RUN/STOP key switch from "RUN" → "STOP" → "RUN" or perform the reset operation with the keys switch.
ERROR	Indicates the CPU module error detection status. On : When a self-diagnosis error that does not stop the operation (except for battery error) is detected. (Set the operation error set mode to "continue" in the parameter mode PC RAS setting.) Off : Normal Flashing : When an error that stops the operation is detected.
USER	The detection status or enunciator F status are indicated for the CHK instruction. On : When an error is detected with the CHK instruction, or when the enunciator is turned on. Off : Normal Flashing : When the latch clear is executed.
BAT.ALARM	Indicates the CPU module and memory battery status. On : When a battery error is due to battery voltage reduction is detected. Off : Normal
BOOT	Indicates the execution status of the boot operation. On : When the execution is complete. Off : When not executed.

(2) The LED that is on can be turned off by the following operation. (Except for the reset operation.)

Method to Turn LED Off	Applicable LED			
	ERROR	USER	BAT.ALARM	BOOT
Execute the LEDR instruction after resolving the cause of error.	○	○	○	×
After the cause of error is resolved, cancel the error by operating the special relay SM50 and special register SD50. (Only for the operation continue errors.) *1	○	○	○	×
Turn off the LED by operating the special relay SM202 and special register SD202. *1	×	○	×	○

○ : Valid × : Invalid

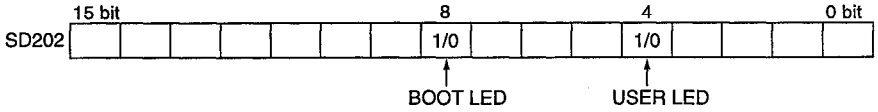
*1: Special relay and special register contents

SM50 . . . . . When switch from OFF to ON, the error is canceled for the error code stored in the SD50.

SD50 . . . . . The error code for the error to be canceled is stored. (Refer to Section 22.3.2.)

SM202 . . . . When turned from OFF to ON, the LED corresponding to each bit in the SD202 are turned off.

SD202 . . . . This specifies the LED to turn off. (Only USER LED and BOOT LED can be turned off.)



1 means "turn off" and 0 means "leave on" in the setting.

The setting to turn off each LED is as follows. (All in hexadecimal.)

- When turning off both LEDs: SD202=110H
- When turning off only the BOOT LED: SD202=100H
- When turning off only the USER: LED:SD202=10H

(3) Method to not display the ERROR LED, USER LED, and BAT. ALARM LED

The ERROR LED, USER LED, and BAT. ALARM LED have the same priorities as in Section 9.8.2, "Priority Setting".

When a error number for an LED is deleted from this priority, the LED will not turn on even if an error with that error number occurs.

(Refer to Points in Section 9.8.2 for the setting method.)

9.8.2 Priority Setting

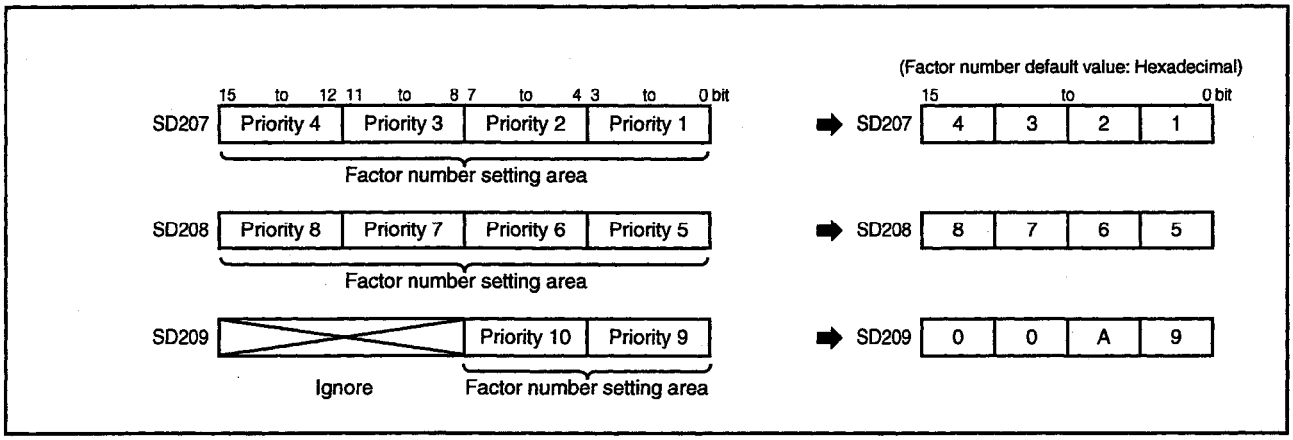
When multiple factors that can be displayed occur, the display is performed with the following conditions:

(1) A stop error is displayed without condition.

(2) An operation continue error is displayed according to the priority factor number set as the default.
The priority can be changed. (Set with the special register SD207 to SD209.)

(3) When errors with the same priority level occur, the error detected first is displayed.

The priority is set with the special registers SD207 to 209 in the following manner:



The description and default priority for the factor number to be set in the special registers SD207 to SD209 are as follows:

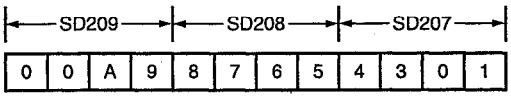
Priority	Factor number (Hexadecimal)	Description	Remarks
1	1	AC DOWN	AC power, DC power shutoff
2	2	UNIT VERIFY ERR. FUSE BREAK OFF SP. UNIT ERROR	I/O module verification error Fuse shutoff Special function module verify error
3	3	OPERATIN ERROR LINK PARA. ERROR SFCP OPE. ERROR SFCP EXE. ERROR	Calculation error Link parameter error SFC instruction calculation error SFC program execution error
4	4	ICM.OPE. ERROR FILE OPE. ERROR EXTEND INST. ERROR	Memory card operation error File access error Extension instruction error
5	5	PRG.TIME OVER	Constant scan setting time time up Low-speed execution monitoring time time up
6	6	CHK instruction	
7	7	Annunciator	
8	8	—	
9	9	BATTERY ERR.	
10	A	Clock data	

POINTS

(1) To keep the LED off at occurrence of the above error, set the factor number setting area, where the corresponding factor number is stored in SD207 - SD209, to 0.

[Example]

To leave the ERROR LED off when a fuse shutoff error is detected, set the factor number setting area to "0" where the error number is "2".



Because the factor number "2" is not set, the ERROR LED remains off even if the fuse shutoff is detected. In this case, even if another error with the factor number "2" (I/O module verify error or special function module verify error) is detected, the ERROR LED remains off.

(2) Even if the LED is set to be turned off, error code storage is performed for SM0 (diagnosis error flag) on, SM1 (self-diagnosis flag) on, and SD0 (CPU diagnosis error register).

10 Other Functions

10.1 Function List

Other functions are listed below:

Item		Description	Reference
Constant scan		Function to execute the program in a set time interval regardless of the program scan time.	Section 10.2
Latch function		Function to maintain the device data when performing the reset operation of the CPU module during PLC power-off.	Section 10.3
Output status setting when STOP → RUN		Function to set the output Y status (output before STOP/output after the calculation execution) when the CPU module is set from STOP status →RUN status.	Section 10.4
Clock function		Function to execute the CPU module internal clock.	Section 10.5
Remote operation		Function to operate the Q2AS from a remote place.	Section 10.6
	Remote RUN/STOP	Function to stop and start operating the CPU module.	Section 10.6.1
	Remote STEP → RUN	Function to perform step-by-step CPU module operation.	Section 10.6.2
	Remote PAUSE	Function to temporarily stop the CPU module.	Section 10.6.3
	Remote RESET	Function to reset the CPU module.	Section 10.6.4
	Remote latch clear	Function to clear the CPU module latch data.	Section 10.6.5
	Relationship between remote operation and CPU key switch	When performing remote operation, the relationship with the CPU module RUN/STOP key switch setting is explained.	Section 10.6.6
Terminal setting		Function uses the display device and key input of the Q6PU programming module.	Section 10.7
	Message display	Function to display a message to the Q6PU display device.	Section 10.7.1
	Key input operation	Function to read the Q6PU key input.	Section 10.7.2
Read the module access interval time		Function to monitor the special function module, network module, and access the interval time for the peripheral device (time from CPU module access receipt to the next access receipt.)	Section 10.8

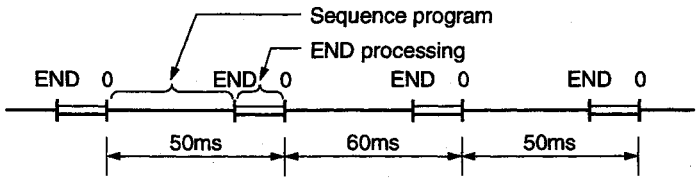
Refer to "GX Developer OPERATING MANUAL" or "SW IVD-GPPQ OPERATING MANUAL (Online)" for the GPP function operation method details.□

10.2 Constant Scan

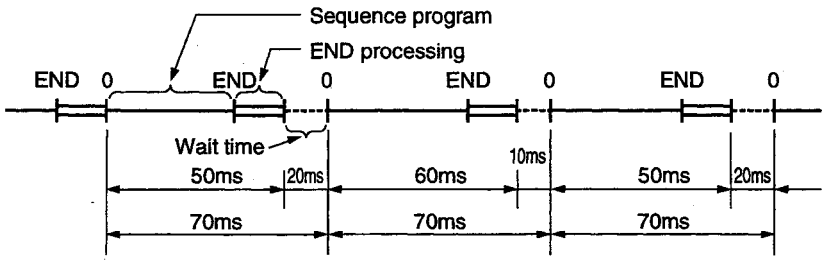
(1) What is Constant Scan?

With the Q2ASCPU, the scan time differs because the processing time differs depending on whether the instruction, which is used in the sequence program, is executed or not. Constant scan is a function to execute the sequence program repeatedly while maintaining the scan time to a constant time.

Scan time when constant scan is not used



Scan time when the constant scan setting is set to 70 ms.



Scan time when constant scan is set to 100 ms during multiple program execution

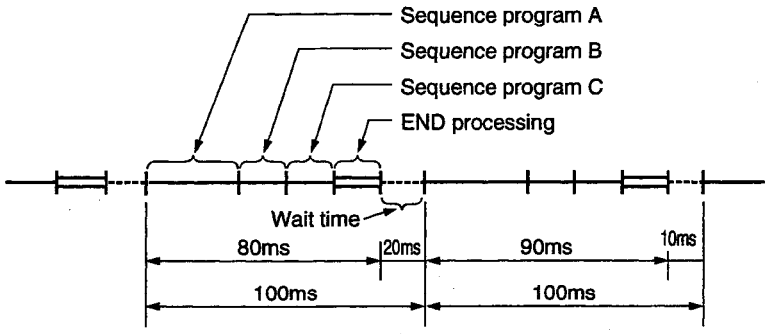


Fig. 10.1 Constant scan operation

When using a low-speed program, the constant scan function setting or low-speed program execution time must be set.

(Refer to "QnACPU PROGRAMMING MANUAL (Fundamentals)" for details.

(2) Setting the constant scan time

(a) The constant scan time setting is performed with the GPP function parameter mode PC RAS setting.

When executing constant scan set, the constant scan.

When not executing a constant scan, set a "blank" to the constant scan time.

[Example]

When the constant scan is set to 100 ms.

PC RAS Setting	
4. Constant Scan	[100] ms

(b) Set the set time of the constant scan larger than the maximum scan time of the sequence program. If the sequence program scan time is larger than the constant scan set value, the Q2ASCPU detects an error code (SD0=5010), the sequence program is executed with the scan time by ignoring the constant scan. Also, set the constant scan set time less than the WDT (Watch Dog Timer) set time. If the value is larger than the WDT set time, the Q2ASCPU detects a WDT error and stops the program execution.

Set the constant scan set time in the following range:

WDT Set Time > Constant Scan Set Time > Sequence Program Maximum Scan Time

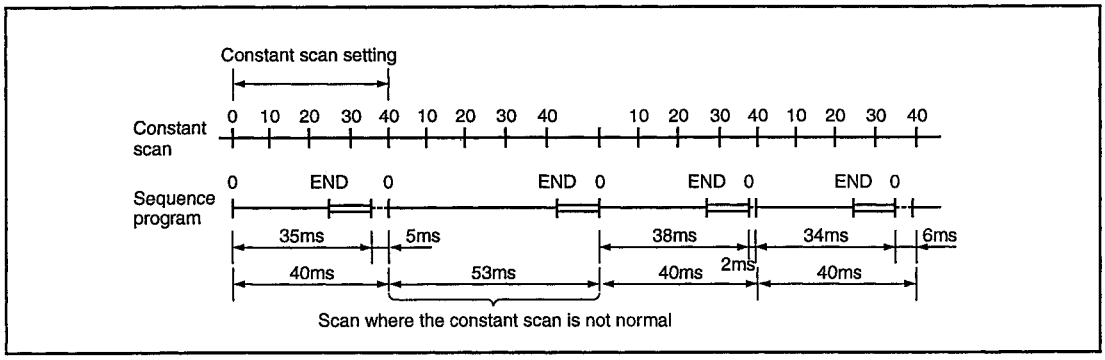


Fig. 10.2 Operation when the Scan Time is More than the Constant Scan

(c) The sequence program processing stops during the wait time from the last END processing execution until the next scan starts.

However, if an interrupt error occurs after the END processing is executed, or a low-speed program exists, the interrupt program or low-speed program is executed.

(d) Constant Scan Time Difference

When a low-speed program exists when a constant scan is being executed, the constant scan time may shift in time up to the following value:

Error = Maximum processing time for one instruction in the low-speed program + Low-speed END processing time

(Time to execute the low-speed program END processing)

The low-speed program executes the program by dividing the constant scan time allowance time. Therefore, if a constant scan is over during an instruction which takes long processing time, the constant scan is complete when the instruction being executed is complete with processing. The time which elapsed until the instruction was complete is the constant scan time difference. Refer to "QCPU(Q mode)/QnACPU PROGRAMMING MANUAL (Common Instructions)" for the command processing time.

10.3 Latch Functions

Each Q2ASCPU device is cleared when the PLC power is turned on, it is reset with the RUN/STOP key switch of the CPU module, and when there is a momentary power failure for more than a permissible amount of time. The values are set back to the default (bit device: OFF and word device: 0).
Latch is a function to maintain the device details when the PLC power is turned on, it is reset with the RUN/STOP key switch of the CPU module, or when there is a momentary power failure for more than a permissible amount of time.

(1) Latch Usage
The latch can be used to continue the control by maintaining the production quantity, defect count, and address even when there is a momentary power failure for more than a permissible amount of time.

(2) Device that can Set Latches
(a) The following devices can use the latch function:

1) Latch relay
2) Link relay
3) Annunciator
4) Edge relay
5) Timer
6) Retentive timer
7) Counter
8) Data register
9) Link register

POINT

Even if the device has specified the latch function, device will not retain the latch if the local device is specified or the device initialization is specified.

(b) The latch range setting is performed with the GPP function parameter mode device setting. There is a range in which the latch clear key becomes valid or invalid in the latch range setting.
Refer to "QnACPU PROGRAMMING MANUAL (Fundamentals)" for each devices latch range.

POINTS

The device details of the latch range is maintained with the battery (A6BAT) attached to the CPU module.

(1) A battery is necessary even if the sequence program is stored as a ROM in a memory card for usage.

(2) The latch range device contents are destroyed when the battery connector is disconnected from the CPU module connector when the PLC power is turned off.

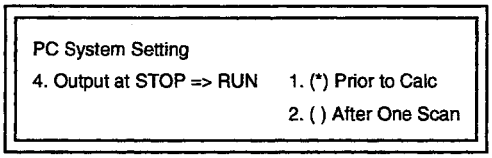
(3) Clearing the Latch Range Device Data

(a) To clear the latch range device contents and set the default values, perform a "latch clear". When latch clear is performed, the device details of devices in the unlatched range will be cleared as well.
However, for the devices set in the parameter in a range where the latch clear key becomes invalid, the device is not cleared even if a latch clear is performed.

(b) Refer ro Section 12.4 for the method to perform latch clear.

10.4 Setting the Output (Y) Status when Changing from STOP Status to RUN Status

When changing from RUN status to STOP status, the RUN status output (Y) is stored in the sequence. Whether to output the output (Y) again prior to calculations, or output after calculation execution is set in the GPP function parameter mode PC system setting.



(a) Prior to Calc . . . . . After the output (Y) status before the STOP status is output, the sequence program calculations are performed.

(b) After One Scan . . . . Clears all output (Y) and outputs the output (Y) after executing the sequence program calculations.

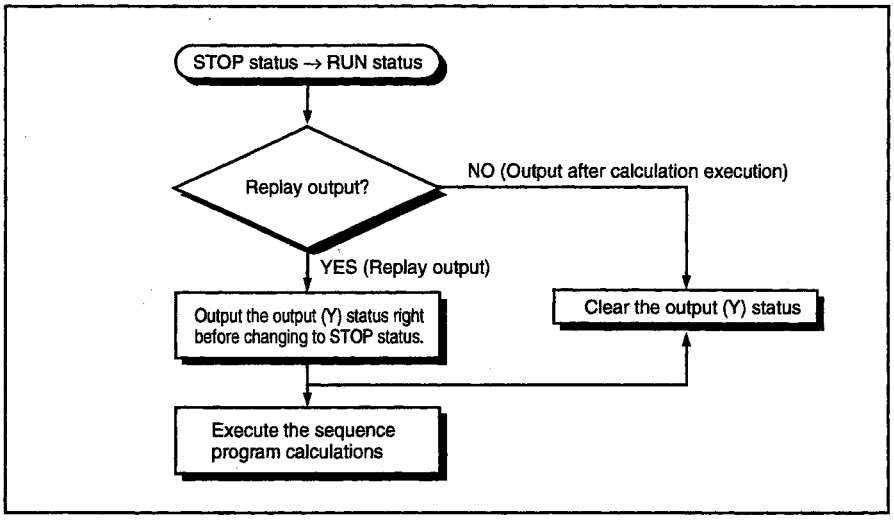

STOP status → RUN status
Replay output?
NO (Output after calculation execution)
YES (Replay output)
Output the output (Y) status right before changing to STOP status.
Clear the output (Y) status
Execute the sequence program calculations

Fig. 10.3 Processing when Change from STOP Status → RUN Status

10.5 Clock Function

The Q2ASCPU has a clock function in the CPU module.
Because the time data from the clock function can be read by the sequence program, the time data can be used for time maintenance of user system.
Also, the time data is used for time maintenance for the CPU module system functions such as for failure history.
The clock operations for the clock function are maintained even when the PC power is off or when there is a momentary power failure for more than the permitted time, using the battery.

POINT

The CPU module system uses the time data for failure history, etc., so set the accurate time when using the CPU module for the first time.

(1) Clock Data

The time data is the year, month, day, hour, minute, second, and weekly day data used for the PC CPU clock element. There are the following:

Data name	Contents	
Year	Two digits in AD	
Month	1 to 12	
Day	1 to 31 (Automatic leap year calculation)	
Hour	0 to 23 (24 hours)	
Minute	0 to 59	
Second	0 to 59	
Weekly Day	0	Sunday
	1	Monday
	2	Tuesday
	3	Wednesday
	4	Thursday
	5	Friday
	6	Saturday

(2) Precision

The precision of the clock function differs from the ambient temperature, as shown below:

Ambient Temperature (°C)	Accuracy (Day difference, S)
0	−1.7 to +4.9s (TYP. + 1.7s)
+25	−1.0 to +5.2s (TYP. + 2.2s)
+55	−7.3 to +2.5s (TYP. + 1.9s)

(3) Writing to the Time Data Clock Element

(a) The following methods can be used to write to the time data clock element.

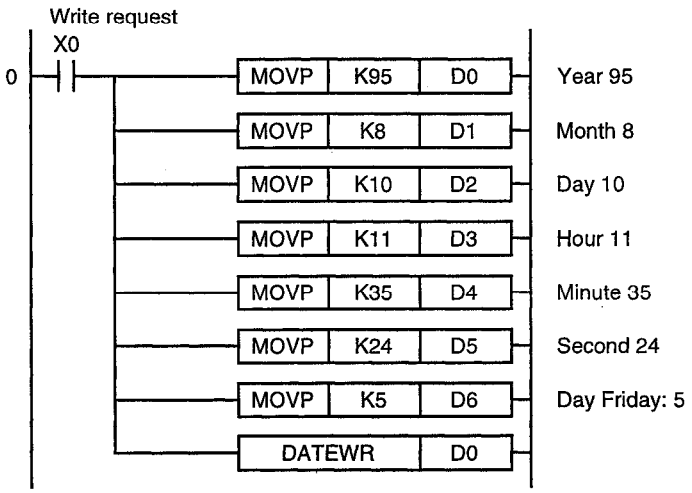
1) Method to Write from the Peripheral Device

- When using the GPP function, the time data is written to the clock element using the clock setting in the PC diagnosis mode PC menu.
- For the Q6PU, the time data is written to the clock element using the clock monitor in another modes PC system monitor.
  (Refer to the operating manual for each peripheral device for details of the peripheral device operations.)

2) Method to Write from the Program

The time data is written in the clock element using the clock instruction (DATEWR).

A program example to write the time data using the time data write instruction (DATEWR).



Refer to "QCPU(Q mode)/QnACPU Programming Manual (Common Instructions)" for details of the DATEWR instruction.

POINTS

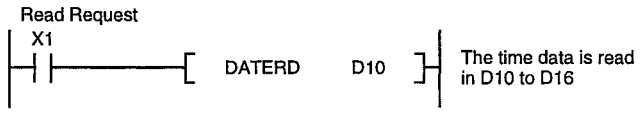
(1) The time data is not written in the clock element at the beginning.
Write the time data once to the clock element when using the Q2ASCPU.

(2) Even if a part of the time data is being corrected, all data must be written to the clock again.

(3) When an invalid time data is written to the clock, normal clock operation cannot be performed.
[Example]
Month 13 Day 32

(4) Reading Time Data

(a) When reading the time data to the data register, use the time data read instruction (DATERD) from the program.
The program example is shown below:



Refer to the "QCPU(Q mode)/QnACPU Programming Manual (Common instructions) for the details of the DATERD instruction.

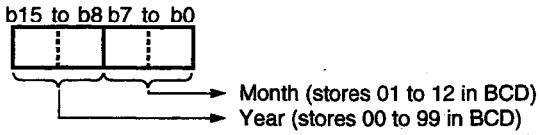
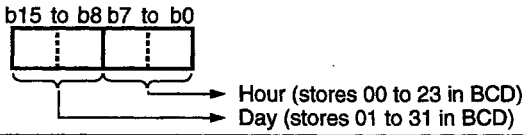
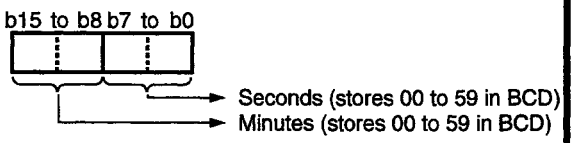
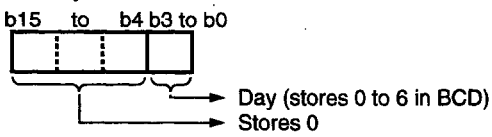
(b) When reading the time data to SD210 to SD213, turn on the SM213 from the sequence program or peripheral device.

(5) Special Relay and Special Register for Time Data Read/Write
The following describes the special relay and special register used for the reading of time data and setting data for clock operations.

(a) The following describes the special relays used for the clock function.

Device	Name	Description
SM210	Clock data set request	• Writes time data to the special registers (SD210 to SD213), and performs clock operations. • After the END instruction is executed for the scan where the SM210 changed from OFF to ON, the time data stored in SD210 to SD213 are written to the clock.
SM211	Clock data error	• Used to determine whether there was an error when the time data was set. • Turns on when each data is not at BCD cord.
SM213	Time data read request	• Used to read time data to the special registers (SD210 to SD213). • When SM213 is on, time data in SD210 to SD213 are read after executing the END instruction is executed.

(b) The following describes the special registers used for the time data:

<table>
<tr><th>Device</th><th>Name</th><th>Details</th></tr>
<tr><td>SD210</td><td>Time data
(Year, Month)</td><td>• The year and month format is as follows. The year is the last 2 digits in AD.
 
</td></tr>
<tr><td>SD211</td><td>Time data
(Day, Hour)</td><td>• The day and hour are as follows:
 
</td></tr>
<tr><td>SD212</td><td>Time data
(Minute, Seconds)</td><td>• The minute and seconds are as follows:
 
</td></tr>
<tr><td>SD213</td><td>Time data
(Day of Week)</td><td>• The day of the week is as follows:
 

• The day of the week setting is as follows:
<table>
<tr><td>Day</td><td>Sunday</td><td>Monday</td><td>Tuesday</td><td>Wednesday</td><td>Thursday</td><td>Friday</td><td>Saturday</td></tr>
<tr><td>Stored data</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr>
</table></td></tr>
</table>

10.6 Remote Operation

The Q2ASCPU can allow control of the CPU module operation status from an external (GPP function, intelligent special function module, and remote contact) source.

Remarks

The serial communication module is used as the example to describe the intelligent special function module.

10.6.1 Remote RUN/STOP

The remote RUN/STOP performs RUN/STOP of the Q2ASCPU from an external source with the CPU module RUN/STOP key switch at RUN.

(1) Remote RUN/STOP Application

Using remote RUN/STOP for the following remote operations are useful:

(a) When the CPU module is at a position out of reach

(b) When performing RUN/STOP of the control board CPU module from an external source

(2) Calculations during Remote RUN/STOP

The program calculation that performs remote RUN/STOP are as follows:

(a) Remote STOP ..... Execute the program to the END instruction and enters the STOP state.

(b) Remote RUN ..... When remote RUN is performed while in the STOP state using remote STOP, the state changes to RUN and executes the program from step 0.

(3) Remote RUN/STOP Method

There are two ways to perform remote RUN/STOP:

(a) Remote RUN contact method

The remote RUN contact is set with the GPP function parameter mode PC system setting.

The device range that can be set is input X0 to 1FFF.

By turning the set remote RUN contact ON/OFF the remote RUN/STOP can be performed.

1) When the remote RUN contact is OFF, the CPU module enters the RUN state.

2) When the remote RUN contact is ON, the CPU module enters the STOP state.

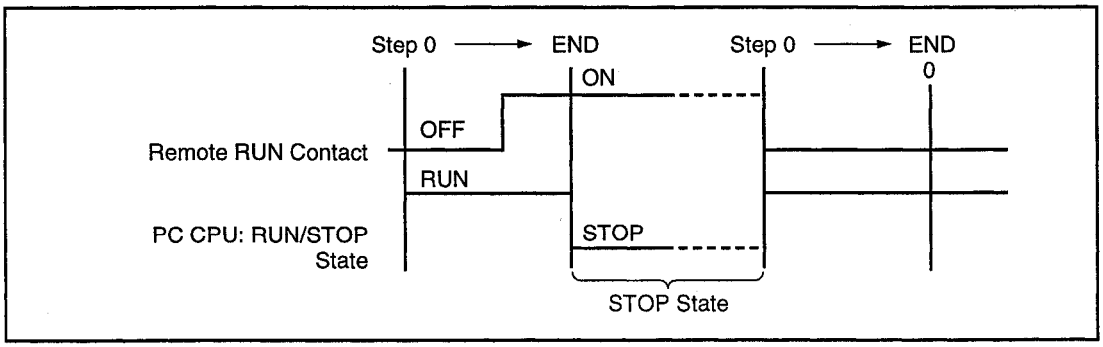


Fig. 10.4 Time Chart for RUN/STOP with Remote RUN Contact

(b) Method using the GPP function, serial communication module, etc.
CPU module RUN/STOP can be performed by the remote RUN/STOP operation from the GPP function, serial communication module, etc.
The GPP function operation is performed with remote operations in each modes PC menu.
The serial communication module control is performed using its own protocol command.
Refer to "AJ71QC24 (-R2/R4) Serial Communications Module USER'S MANUAL" for details of the serial communication module control.

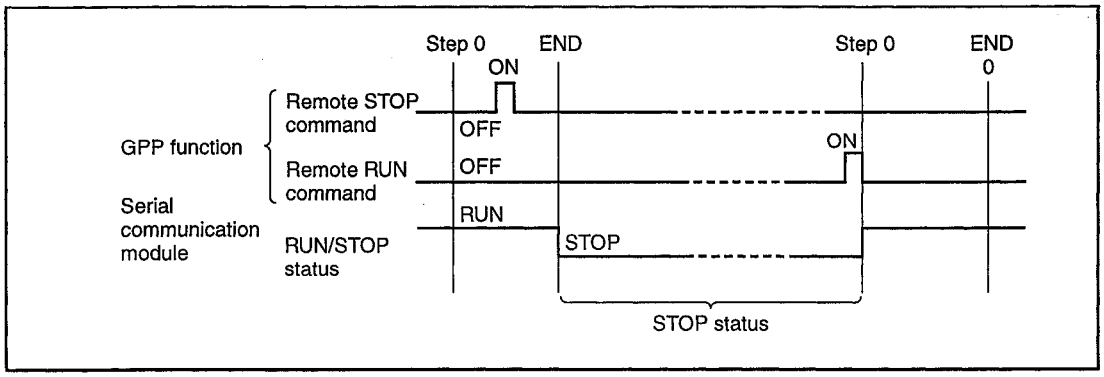


Fig. 10.5 Remote RUN/STOP Time Chart using GPP function, Serial Communication Module, Etc.

(4) Precaution

(a) Take note of the following, because STOP has priority in Q2ASCPU:

1) The Q2ASCPU enters the STOP state when remote STOP is performed from remote RUN contact, GPP function, or serial communication module.

2) When Q2ASCPU is set to the STOP state with remote STOP, all external factors which performed a remote STOP (remote RUN contact, remote communication module, etc.) must be set to RUN.

Remarks

The RUN/STOP state is described below:

- RUN State . . . . . State which repeatedly executes the calculations from step 0 to END instruction in the sequence program.
- STOP State . . . . . State where the sequence program calculations are stopped and the output(Y) is all OFF.

10.6.2 Remote STEP-RUN

The remote STEP-RUN performs step-by-step operation of the Q2ASCPU from the GPP function, with the CPU module RUN/STOP key switch at RUN position.
Step-by-step operation means that the program calculations are performed step by step from the specified step.
Refer to Section 8.7 for details of step-by-step operation.

(1) Remote STEP-RUN Application
The program can be executed by checking the program execution status or device details during system debugging.

(2) Remote STEP-RUN Method
The remote STEP-RUN is performed in the following manner:
(a) Set the CPU module RUN/STOP key switch to RUN.
(b) Perform remote STEP-RUN operation from GPP function.

10.6.3 Remote PAUSE

Remote PAUSE performs the Q2ASCPU PAUSE function from an external source, with the CPU module RUN/STOP key switch at RUN position.
The PAUSE function stops the CPU module calculations while maintaining the ON/OFF state of all output (Y).

(1) Remote PAUSE Application
This can be used to maintain the output (Y) on even if the CPU module is changed to the STOP state, in such areas as process control.

(2) Remote PAUSE Method
There are two ways to use remote PAUSE:

(a) Remote PAUSE Contact Method
The remote PUASE contact is set in the GPP function parameter mode PC system setting.
The device range that can be set is input X0 to 1FFF.

1) The PAUSE state contact (SM204) is turned on when the END processing is executed for the scan with both remote PAUSE contact and PAUSE permission flag (SM206) on.
When the scan after the PAUSE state contact is turned on is executed to the END instruction, the CPU enters the PAUSE state and stops the calculations.

2) When the remote PAUSE contact is off or SM206 is turned off with the GPP function, the PAUSE state is canceled, and the sequence program calculation is performed again from step 0.

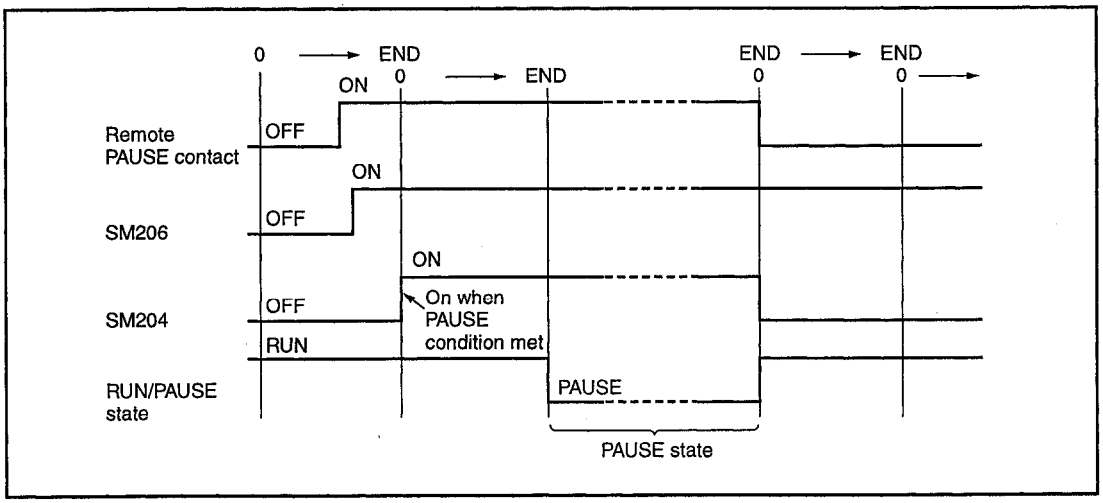


Fig. 10.6 PAUSE Time Chart with Remote PAUSE Contact

Remarks

If the remote RUN contact and remote PAUSE contact are identical, the remote PAUSE contact is made invalid.

(b) GPP function, Serial Communication Module Method

The remote PAUSE operation can be performed from the GPP function or serial communication module.

The GPP function operation is performed from each modes PC menu remote operation.

The serial communication module control is performed using its own protocol commands.

Refer to :AJ71QC24 (-R2/R4) Serial Communications Module USER'S MANUAL" for details of the serial communication module control.

1) When the END processing is executed for the scan where the remote PAUSE command was accepted by the GPP function, the PAUSE state contact (SM204) is turned on.
   When the scan after the PAUSE state contact is turned on is executed to the END process, it enters the PAUSE state and stops the calculations.

2) When the remote RUN command is received from the GPP function, the sequence program calculations are performed again from step 0.

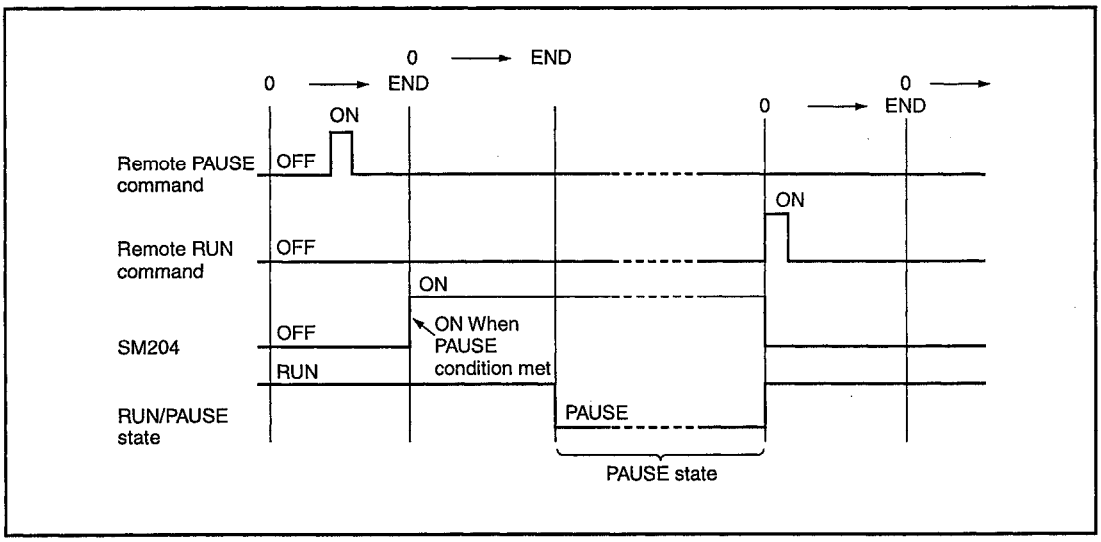
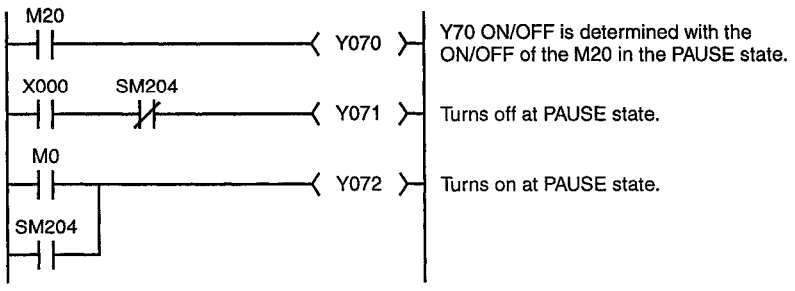


Fig. 10.7 PAUSE Time Chart with GPP function

POINT

To set the output (Y) ON/OFF status when change to the PAUSE state, perform an interlock with the PAUSE state contact (SM204).



10.6.4 Remote RESET

The remote RESET resets the Q2ASCPU from an external source when the CPU module is at STOP state.
Even if the CPU module RUN/STOP key switch is at RUN, the reset can be performed when the CPU module is stopped and an error that can be detected by the self-diagnosis function occurs.

POINT

Remote RESET cannot be performed when the CPU module is at RUN state.

(1) Remote RESET Application
Remote RESET can reset the CPU module remotely when an error occurs for which the CPU module cannot be operated directly.

(2) Remote RESET Method
The remote RESET can only be performed from the GPP function or serial communication module operation.

(a) The remote RESET must be enabled with the parameters before performing a reset operation from a GPP function or serial communication module. The remote RESET enable/disable is set at GPP function parameter mode PC system setting.

(b) After enabling the remote RESET in the parameter, the reset is performed with the remote operation.
- For the GPP function, this is performed in each modes PC menu remote operation.
- For the serial communication module, its own protocol command is used.

Refer to "AJ71QC24(-R2/R4) Serial Communications Module USER'S MANUAL" for the details of the serial communication module control.

10.6.5 Remote Latch Clear

The remote latch clear resets the device data latched to the Q2ASCPU using the GPP function, when the CPU module is at the STOP state.

POINT

Remote latch clear cannot be performed when the CPU module is at RUN state.

(1) Remote Latch Clear Application Usage

Remote latch clear is useful when the CPU module is in the following areas. In these cases, the operations are performed in combination with the remote RUN/STOP.

- When the CPU module is at a position out of reach
- When performing latch clear of the control board CPU module from an external source

(2) Remote Latch Clear Method

The remote latch clear can only be performed from GPP function or serial communication module.

- The GPP function operations are performed with each modes PC menu remote operation.
- The serial communication module control uses its own protocol commands.

Refer to "AJ71QC24 (-R2/R4) Serial Communications Module USER'S MANUAL" for details of the serial communication module control.

POINTS

(1) The latch range for the device set in the parameter mode device setting has a range that makes latch clear valid or invalid. The remote latch clear is valid only for devices set in the range "Enable C/L key"

(2) Devices that are not latched are cleared when the remote latch clear is executed.

10.6.6 Relationship of the Remote Operation and RUN/STOP Key Switch of CPU Module

The Q2ASCPU operation status is as follows with the combination of remote operations in Sections 10.6.1 to 10.6.5 and RUN/STOP key switch of CPU module.

Remote operation / Key switch	RUN *1	STEP-RUN	STOP	PAUSE *2	RESET *3	Latch clear
RUN	RUN	STEP-RUN	STOP	PAUSE	Cannot operate*4	Cannot operate*4
STOP	STOP	STOP	STOP	STOP	RESET	Latch clear

*1 When performing the operation with remote RUN contact, "RUN-PAUSE contact" must be set in the parameter mode PC system setting.

*2 When performing the operation with remote PAUSE contact, "RUN-PAUSE contact" must be set in the parameter mode PC system setting.
In addition, the remote PAUSE enable coil (SM206) must be set ON.

*3 "Remote reset enable" must be set in the parameter mode PC system setting.

*4 RESET can be performed if the CPU module changed to the STOP state from a remote operation.

When multiple remote operation requests occur when the RUN/STOP key which is at RUN, the CPU module performs the operation with higher priority first.

Remote operation	RUN	STEP-RUN	STOP	PAUSE	RESET	Latch clear
Priority	4	3	1	2	—	—

The priority increases from 4 to 1.

10.7 Terminal Operation

This function performs the following data communication using the Q2ASCPU peripheral device instructions with the Q6PU programming module at terminal mode.

(1) Displays the messages from the Q2ASCPU to the Q6PU display device.

(2) Stores the Q6PU key entry to the Q2ASCPU device.

As seen above, Q6PU can be used as the Q2ASCPU terminal.

Each function is described in the next section. Refer to "QCPU(Q mode)/QnACPU PROGRAMMING MANUAL (Common Instructions)" for the peripheral device instruction.

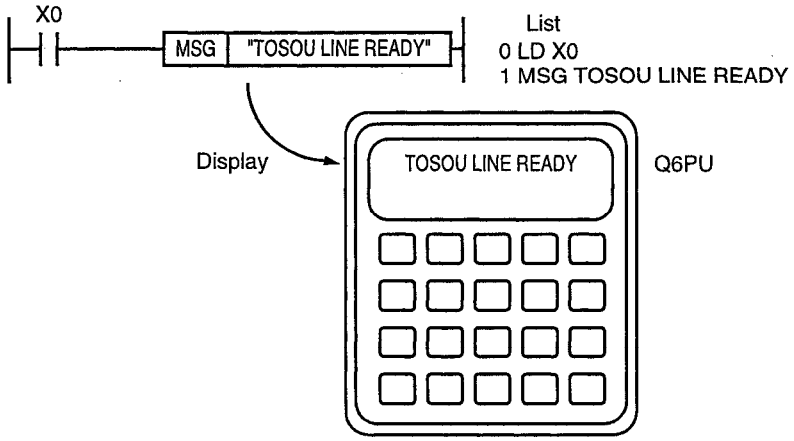
10.7.1 Message Display Operation

The specified character string is displayed in the Q6PU with the peripheral device instruction "MSG."

Even for the GPP function, the character string can be displayed from the CPU message in the PC diagnosis mode display menu.

[Example]

Program to display "TOSOU LINE READY" as Q6PU message No. 1 when X0 is turned on.



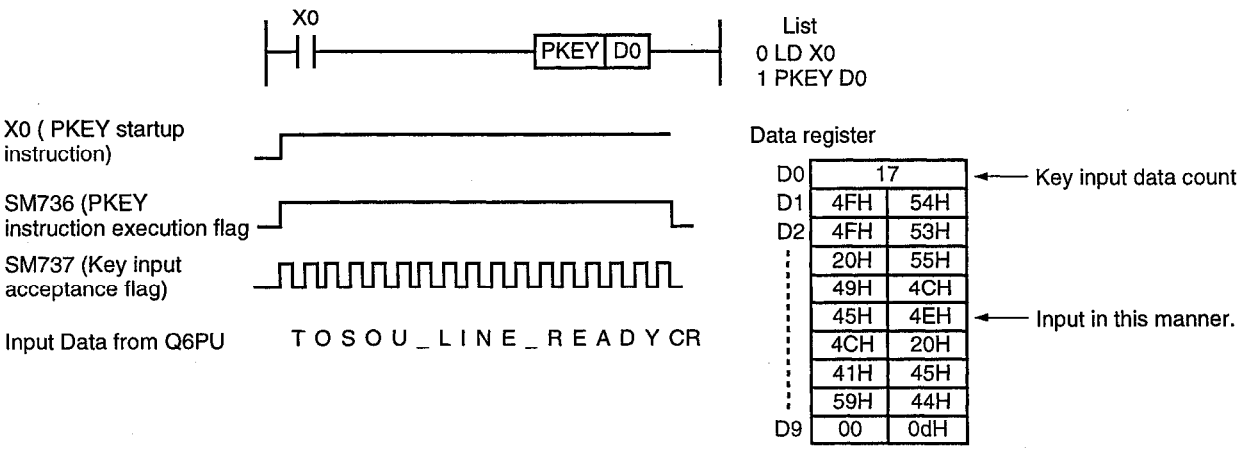
10.7.2 Key Entry Operation

With the peripheral device instruction "PKEY," the character data entered with the Q6PU is stored in the specified device as ASCII data.

The data entry completion is when the CR code is entered or when the 32nd character is entered.

[Example]

A program that enters "TOSOU LINE READY" on the Q6PU when the X0 is turned on.



10.8 Reading the Module Access Time Interval

The Q2ASCPU can monitor the access time interval (time from an access receive and the next access receive) of the intelligent special function module, network module, data link module, or GPP function. The access frequency to the CPU module from an external source can be found out.

To read the module access time interval, the next special relays and special registers are performed.

(1) Special Relay

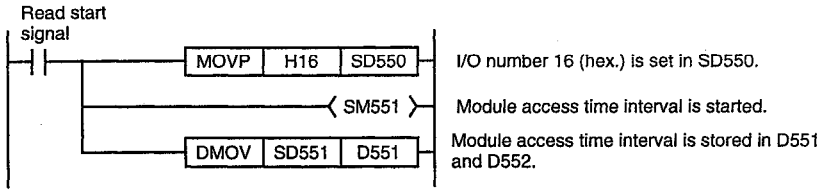
Number	Name	Description
SM551	Module access time interval read	When turned on, the module access time interval from the special function module specified with the special register SD550 is read to the SD551 to SD552. ON : Read OFF : Not processed

(2) Special register

Number	Name	Description
SD550	Module access time interval measurement module	The I/O number of the module to measure the module access time interval is set. Set the I/O number for the peripheral device to be connected to the CPU module RS-422 interval to HFFFF. The I/O number is set by deleting the lower digit.
SD551 to SD552	Module access time interval	When SM551 is turned on, the access time interval for the module specified in SD550 stored. SD551: 1ms unit (range 0 to 65535) SD552: 1 μs unit (range 0 to 900, stored every 100 μs) [Ex.] Module Access time interval For 123.4 ms SD551=123, SD552=400

[Program Example]

When reading the module access time interval for the X/Y160 special function module.



POINT

To read the access time interval accessed from another GPP function on the network, set the network modules I/O number.

Remarks

The module access time interval indicates the interval of a transient request for monitor, test, program read/write, etc.

The access time interval during cyclic communication from the network or data link module is not stored.

11 Comments That Can Be Stored in Q2ASCPU

11.1 Function List

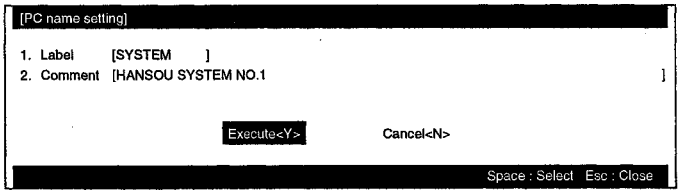
The Q2ASCPU can store various comments. By doing this, the CPU module operation is improved. Also, this helps the readability of the program for other program designers.
The various comment list that can be stored in the Q2ASCPU is shown below:

Item	Function	Reference
PC name	Names the CPU module to be used.	Section 11.2
Drive read title	Attaches the header.	Section 11.3
File read title	Attaches the header for each file.	Section 11.4
Device comment	Attaches the comments and labels for the devices used in the program.	Section 11.5
Statement/Note	Attaches the commenst for the step number, P, or I pointer from the program.	Section 11.6
Device initialization comment	Attaches comments to the device initialization file.	Section 11.7

Refer to "GX Developer OPERATING MANUAL" or "SW □IVD-GPPQ OPERATING MANUAL (Offline)" for details of the setting method for each function.

11.2 Name

The PC name simplifies the CPU module checking when the Q2ASCPU is accessed by the GPP function, placing the comments to the CPU module.
The PC names set the two types of labels and comments. The setting is performed in the GPP function parameter mode "PC name setting" screen.

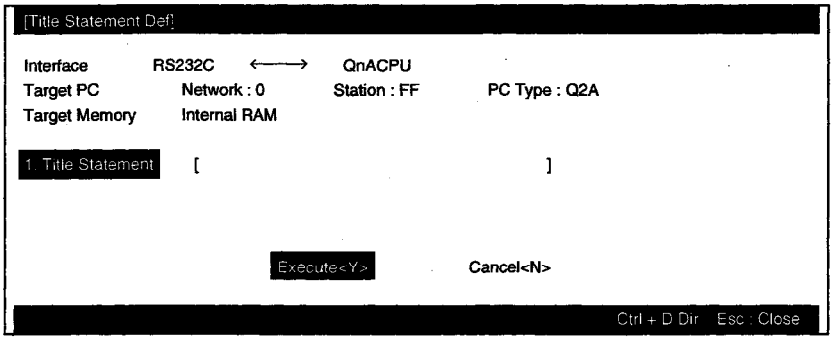


The setting details are as follows:

Item	Setting details	Setting range	Default value
Label	CPU module label is set.	Max. 10 char.	No setting
Comments	CPU module comment is set.	Max. 64 char.	

11.3 Driver Title

The drive title is placed to identify the files in the internal RAM and memory card.
The drive header is created in the "Title Statement Def" in the GPP function online mode PC menu.

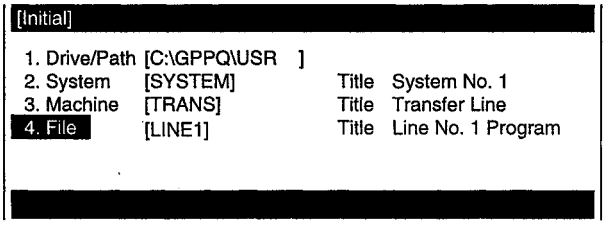

[Title Statement Def]
Interface RS232C ←——→ QnACPU
Target PC Network : 0 Station : FF PC Type : Q2A
Target Memory Internal RAM
1. Title Statement [ ]
Execute<Y>
Cancel<N>
Ctrl + D Dir Esc : Close

POINT

The drive title uses one file worth of memory in each memory when created.

11.4 File Title

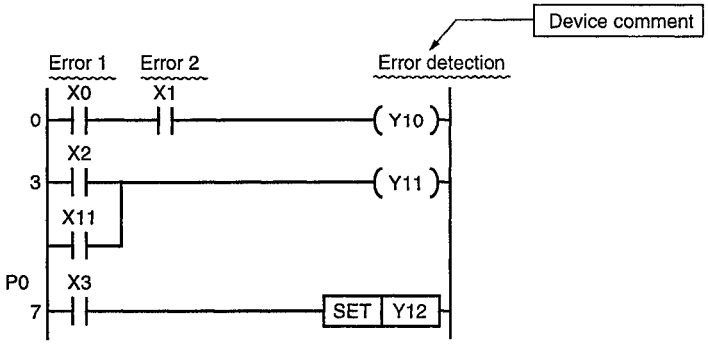
The file title is placed to identify the file contents. The file title is set when setting the file at GPP function startup or when in the PC write in the PC menu of each mode. A maximum of 32 characters can be stored.


[Initial]
1. Drive/Path [C:\GPPQ\USR]
2. System [SYSTEM] Title System No. 1
3. Machine [TRANS] Title Transfer Line
4. File [LINE1] Title Line No. 1 Program

The file title is stored in each file.
Note that the file title is not stored into the file register file.

11.5 Device Comments

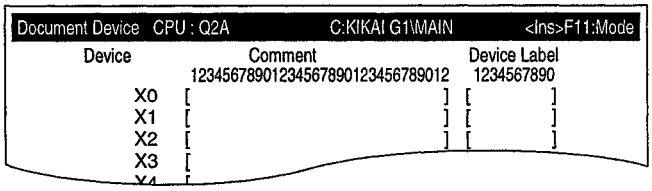
The device comment attaches comments to the device to improve readability by displaying the comment to the device.
The program can be created instead of device names by setting "general" as the CPU type when creating the program with the GPP function.



POINT

A memory card is necessary to store the device comment file to the CPU module when the device comments are created.

(1) The device comment setting is performed in the GPP function text editor mode.
A maximum of 32 characters can be set for comments and up to 10 characters for the label (device name).

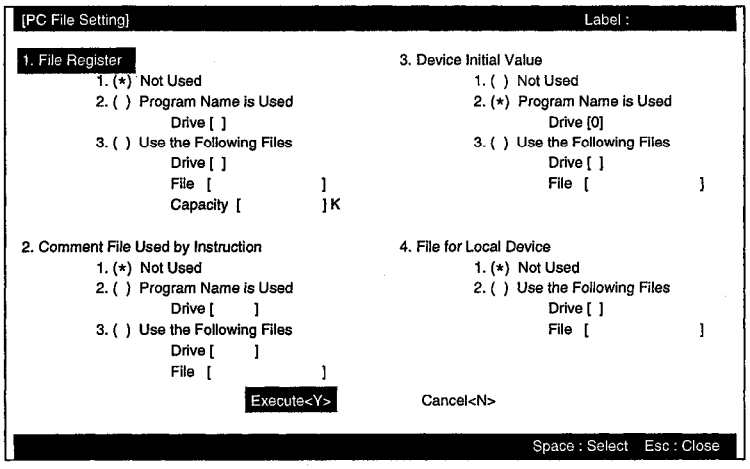


- The devices that can attach comments and labels are as follows:

  Device name : X, Y, M, L , F, SM, B, SB, V, T (current value), C (current value), ST (current value), D, SD, W, SW, R, ZR, P, I, U□\□G, J□\X, J□\Y, J□\B, J□\SB, J□\W, J□\SW, BL□\S, and BL□\TR.

  (If the P, I comments are used as the subroutine program or interrupt program pointers, the comments are not displayed. To display the comments, display them as pointer statements. (Refer to Section 11.6.))

(2) When using comments for the application instructions (LEDC, PRC, etc.) and the device comment file is written, the device comment file to be valid is set in the parameter.
The setting is performed in the GPP function parameter mode "2. Comment File Used by Instruction" in "PC File Setting" screen.


[PC File Setting]
Label :
1. File Register
1. (*) Not Used
2. ( ) Program Name is Used
Drive [ ]
3. ( ) Use the Following Files
Drive [ ]
File [ ]
Capacity [ ]K
2. Comment File Used by Instruction
1. (*) Not Used
2. ( ) Program Name is Used
Drive [ ]
3. ( ) Use the Following Files
Drive [ ]
File [ ]
3. Device Initial Value
1. ( ) Not Used
2. (*) Program Name is Used
Drive [0]
3. ( ) Use the Following Files
Drive [ ]
File [ ]
4. File for Local Device
1. (*) Not Used
2. ( ) Use the Following Files
Drive [ ]
File [ ]
Execute<Y>
Cancel<N>
Space : Select Esc : Close

The setting details is as follows:

1) Not set by the parameter.
The comment file to be used is not set. To use a comment file, the QCDSET instruction is used. (Refer to "QCPU(Q mode)/QnACPU PROGRAMMING MANUAL (Common Instructions)" for details of the QCDSET instruction.)

2) The specified device comment file is used:
The file names stored in the drive specified by the parameter is used.

3) The same file as the program is used:
The comment file with the same file name as the one in the program file is used. (Must exist in the specified drive.)
When the program is switched, the comment file is changed as well.

POINT

(1) When using the QCDSET instruction, take note of the following:

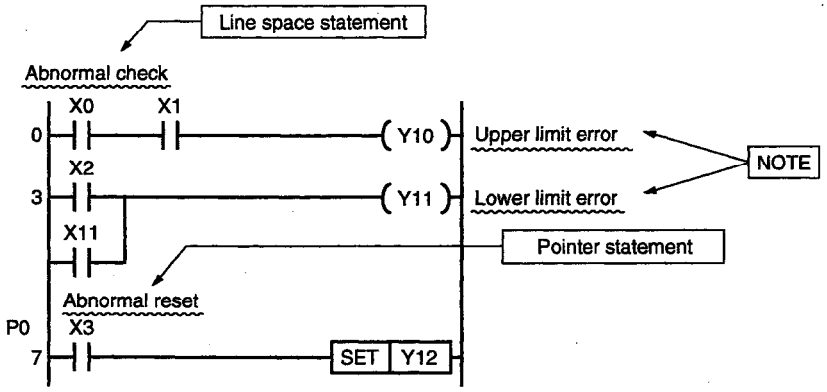
a) When set to 1) or 2) described above, the files set with the QCDSET instruction all become valid.

b) When set to 3), the files set by the QCDSET instruction only becomes valid in the program file that executed the QCDSET instruction.

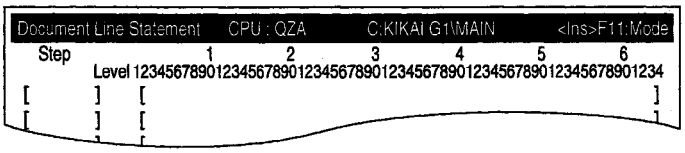
(2) Even if the file set in the parameter does not exist in the specified drive, the CPU module does not yield an error. Also, the file does not exist, so the CPU module does not display comments.

11.6 Statement/Documentation

The statement/documentation improve readability by being attached to the program step or P,I pointer.



(1) The statement/note setting is performed in the "Pointer Statement", "Line Statement", "Document Line Statement", or "Documentation" in the GPP function text editor mode edit menu.



(2) The application for each comment is as follows:

(a) Document Line Statement

In a circuit block grouped by functions, the comments to describe the meaning or application of the function can be attached.

(b) Pointer Statement

The comments to describe the meaning or application of each program can be attached to the pointer of the subroutine program or interrupt program header.

(c) Documentation

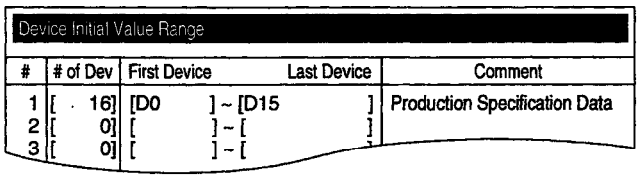
Comments to describe the meaning or application for the function can be placed for each circuit block.

11.7 Device Initialization Comment

The device initialization comment enables the identification of the file by placing comments in the device initialization file.

The device initialization comments are stored in the device initialization file.

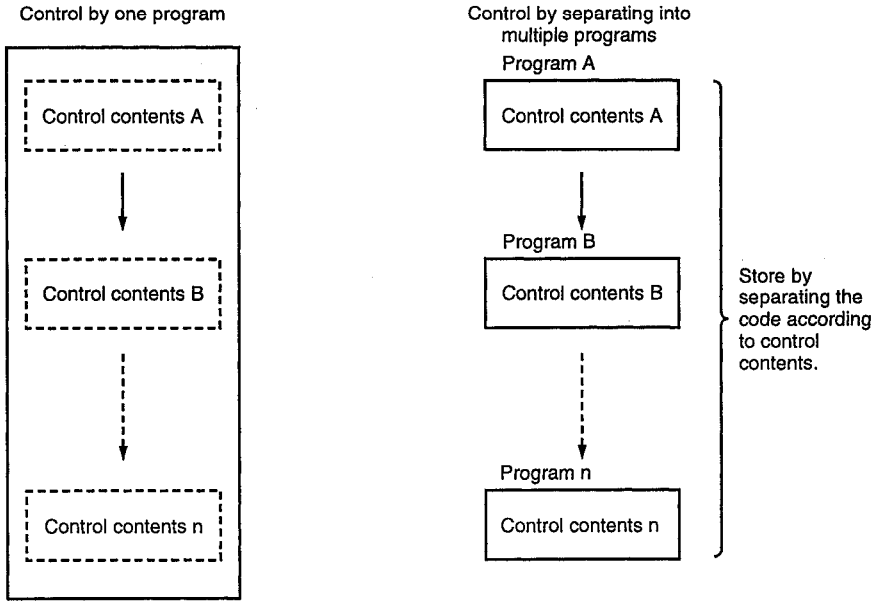
The setting is performed in the "Device Initial Value Range" screen in the GPP function device mode edit menu.



12 Overview of the Q2ASCPU Calculation Processing

12.1 Program Execution Type Overview

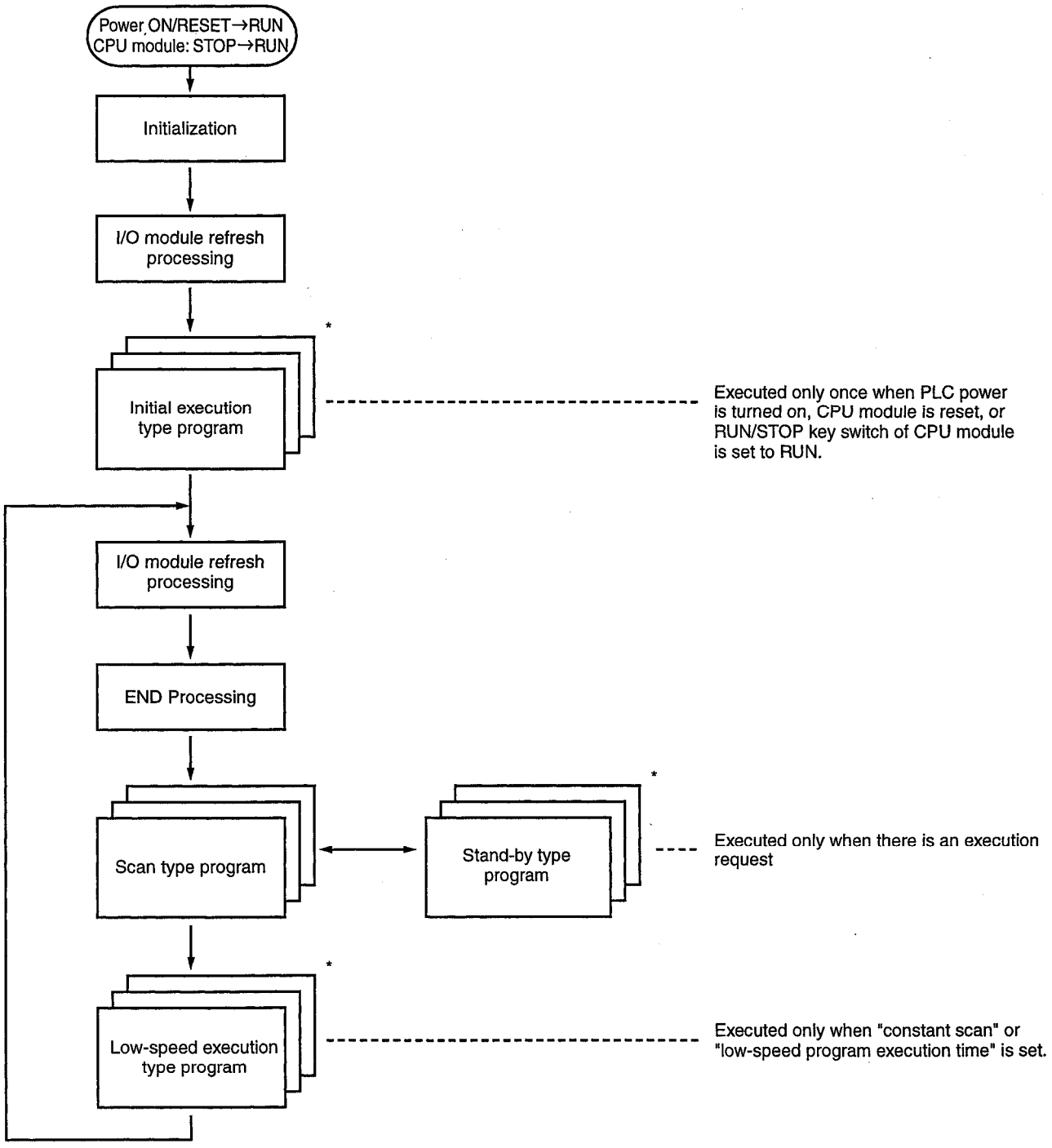
The program executed with the Q2ASCPU is stored in the CPU module internal RAM or memory card.
The internal RAM or memory card can store one program at once, but also can store them as multiple programs by separating them into control units.
Therefore, to create a program with several engineers, each engineer programs each processing unit, and can be stored in the CPU module internal RAM or memory card.



When separating into multiple programs, the "execution type" is set in the GPP function parameter modes program setting.
Each program is executed in the order of setting according to the execution type set in the Q2ASCPU.
There are four types of execution types: "initial execution type," "scan execution type," "low-speed execution type," "stand-by type."

- Initial Execution Type : Program executed only once when the PLC power is turned on, the CPU module is reset or when the RUN/STOP key switch is moved to RUN from STOP. (Refer to Section 12.1.1)
- Scan Execution Type : Program is executed each time a scan takes place from the next scan after the initial execution type program is executed. (Refer to Section 12.1.2)
- Low-Speed Execution Type : Program executed only at scan execution type programs extra time when setting the constant scan or low-speed type program execution time. (Refer to Section 12.1.3)
- Stand-by Type : Program that is executed only when the execution request matches. (Refer to Section 12.1.4.)

The operation processing flow when the PLC power is turned on, the CPU module is reset or the RUN/STOP key switch of the CPU module is set to RUN from STOP is as follows:


Power ON/RESET→RUN
CPU module: STOP→RUN
Initialization
I/O module refresh processing
*
Initial execution type program
Executed only once when PLC power is turned on, CPU module is reset, or RUN/STOP key switch of CPU module is set to RUN.
I/O module refresh processing
END Processing
Scan type program
*
Stand-by type program
Executed only when there is an execution request
*
Low-speed execution type program
Executed only when "constant scan" or "low-speed program execution time" is set.

POINT

(1) Not all execution types need to be set for Q2ASCPU.

Use the items marked with "*" as needed, such as the initial execution type program, low-speed execution type program, and stand-by type program.

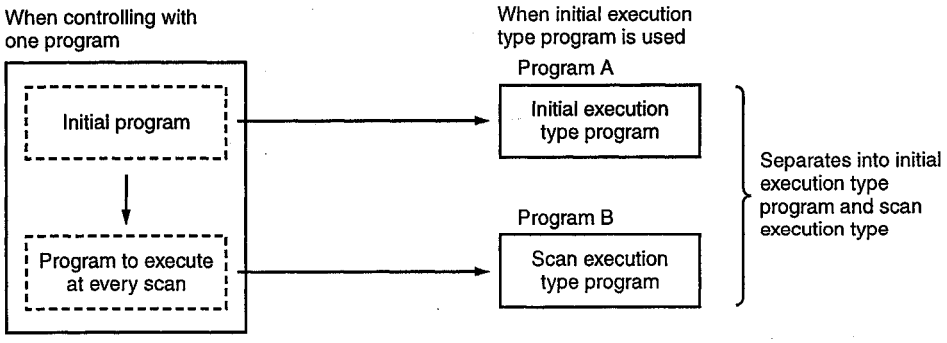
12.1.1 Initial Execution Type Program

(1) What is an Initial Execution Type Program?

(a) The initial execution type program only executes the program once when the PLC power is turned on, when the CPU module is reset, or when the RUN/STOP key switch of the CPU module is moved from STOP to RUN.

(b) The execution type is set to "initialization" in the GPP function parameter modes program setting.

(c) The initial execution type program can be used for programs that do not need to be executed from the next scan after executing once, like the initialization of the special function module.

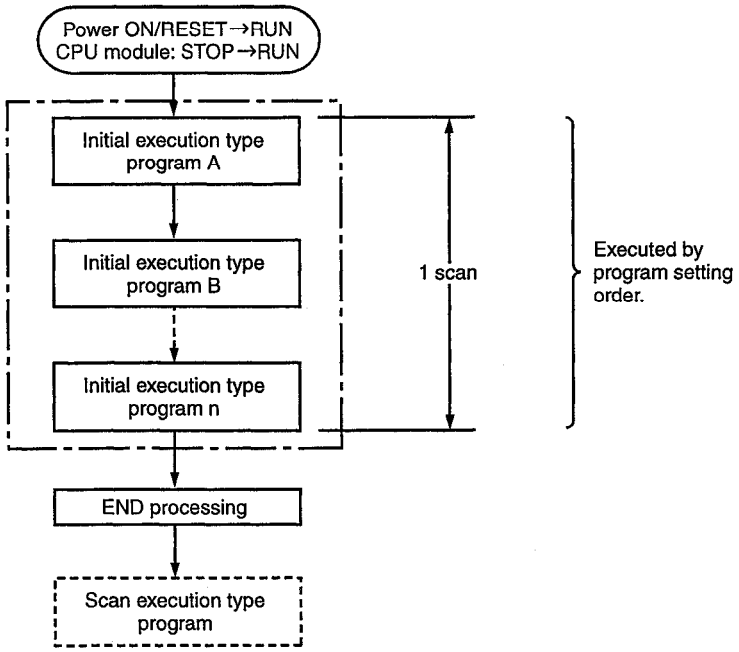

When controlling with one program
Initial program
Program to execute at every scan
When initial execution type program is used
Program A
Initial execution type program
Program B
Scan execution type program
Separates into initial execution type program and scan execution type

(2) Multiple Initial Execution Type Program Execution

If a multiple initial execution type program exists, the parameter modes program is executed in order of setting.

(3) END Processing

When all initial execution type program execution is complete, END processing is performed and the "scan execution type program" is performed from the next scan.



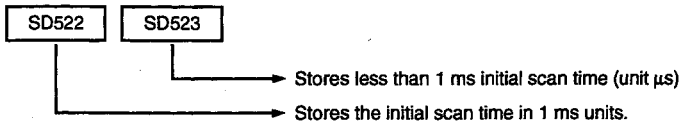
POINT

(1) *: The initial execution type program cannot use an instruction with the complete device.

(4) Initial Scan Time

(a) This is the execution time of the initial execution type program.
When multiple initial execution type program is executed, it is the time until all initial execution type programs are executed.

(b) The Q2ASCPU measures the initial scan time and stores it to the special registers (SD522, SD523).
By monitoring the SD522 and SD523, the initial scan time can be checked.



[Example]
When 3 is stored in SD522 and 400 is stored in SD523, the initial scan time is 3.4 ms.

(5) Initial Execution Monitoring Time
This is a timer to monitor the initial execution type programs execution time. The default value is not set.
When monitoring the execution time of the initial execution type program, it can be set in the range of 10 ms to 2000 ms in the parameter mode "PC RAS Setting". (Setting unit: 10 ms)
When the initial scan time exceeds the set initial execution monitoring time, it is set to "WDT ERROR" and the Q2ASCPU stops operations.

POINTS

(1) *1: The precision of each scan time stored in the special register is ± 0.1 ms.
The initial scan time measurement continues even if the watchdog timer reset instruction (WDT) is executed in the sequence program.

(2) There will be an error in the range of 0 to 10 ms in the measured initial execution monitoring time.
Therefore, when the initial execution monitoring time (t) is set to 10 ms, the initial scan time becomes a WDT ERROR when the initial scan time is in the range 10 ms ≤ t < 20 ms.

12.1.2 Scan Execution Type Program

(1) What is a scan execution type program?

(a) This is a program to be executed once from the next scan after executing the initial execution type program.

(b) Set the execution type to "scan" in the GPP function parameter modes program setting.

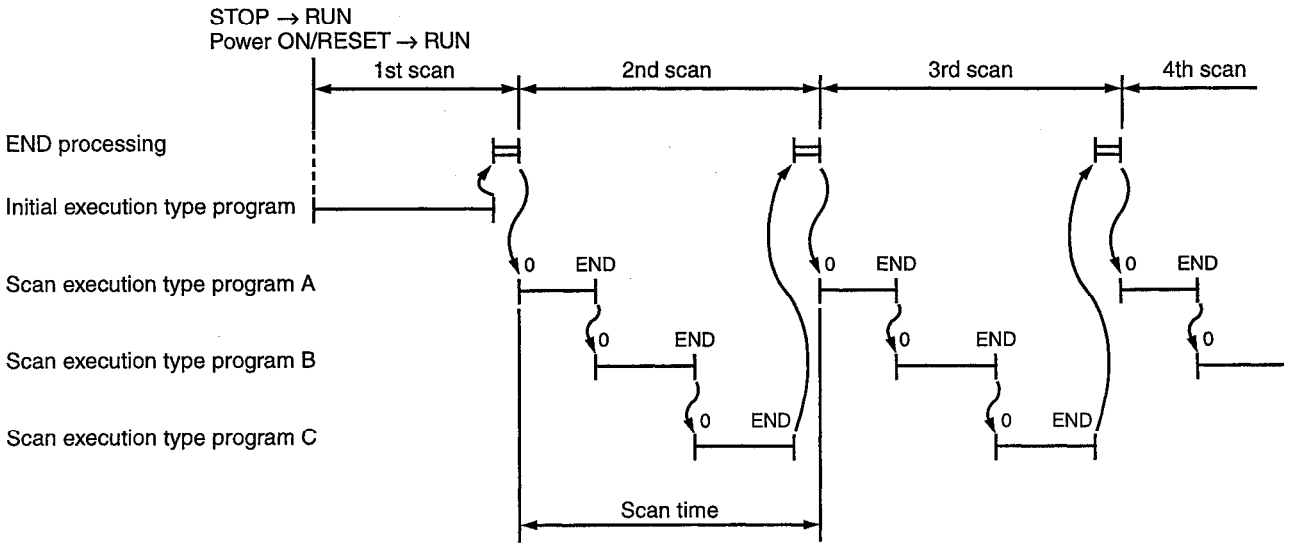
(2) Executing the Multiple Scan Execution Type Program

When there is a multiple scan execution type program, the programs are executed in the order of the small parameter modes program setting number.

(3) END Processing

When all scan execution type programs are executed, the END processing is performed and the first scan execution type program is executed again.

When the COM instruction is placed at the end of the scan execution type program, the END processing can be performed for every program (general data processing and link refresh).



(4) When setting the constant scan *1

When the constant scan is set the scan execution program is executed for every constant scan time set.

Remarks

(1) *1: Constant scan executes the scan execution type program repeatedly in a constant interval.

Refer to Section 10.2 for details.

POINT

(1) Refer to "QnACPU PROGRAMMING MANUAL (Fundamentals)" for index register processing when executing the interrupt program while executing the scan execution type program.

(5) Scan Time

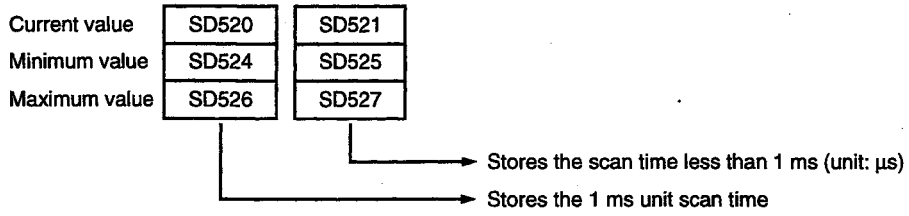
Scan time is a total of following the execution time of the scan execution type program, END processing time, (and low-speed program execution time or constant scant stand-by time) *1.

When a multiple scan execution type is executed, the execution time of the scan execution type program is the time until all scan execution type program execution is complete.

*1: Refer to Section 12.1.3.

- The Q2ASCPU measures current value, minimum value, and maximum value for the scan time, and stores them in the special registers (SD520, SD521, and SD524 to SD527).

By monitoring SD520, SD521, and SD524 to SD527, the scan time can be checked.



[Example]

When 3 is stored in SD520 and 400 is stored in SD521, the scan time is 3.4 ms.

(6) WDT (Watchdog Timer)

This is a timer to monitor the scan time, and the default value is set at 200 ms.

WDT is set in the parameter mode "PC RAS Setting" in the range of 10 ms to 2000 ms. (Setting unit: 10 ms)

When using a low-speed execution type program, set the WDT greater than the added value of the scan time and low-speed execution type program execution time.

When scan time (total of scan execution type program, low-speed execution type program execution time, END processing time, and low-speed END processing time) exceeds the WDT setting time, it becomes a "WDT ERROR" and stops Q2ASCPU operations.

POINTS

(1) *2: The precision of each scan time stored in the special register is ± 0.1 ms.

The initial scan time measurement continues even if the watchdog timer reset instruction (WDT) is executed in the sequence program.

(2) There will be an error of 0 to 10 ms in WDT measurement.

Therefore, when WDT (t) is set to 10 ms, the scan time may not become a WDT ERROR when the initial scan time is in the range of 10 ms ≤ t < 20 ms.

12.1.3 Low-Speed Execution Type Program

(1) What is a low-speed execution type program?

(a) The low-speed execution type program is a program to be executed only for the "constant scan allowance time" or the set "low-speed execution program execution time".

- When setting a constant scan time and prioritize the control precision, the constant scan time is set with the GPP function parameter mode "PC RAS Setting". (Setting range: 5 to 2000 ms, setting unit: 5 ms)
- To perform an accurate low-speed execution type program while maintaining the low-speed execution type programs execution time for every scan, set the low-speed program execution time with the parameter mode "PC RAS setting". (Setting range: 1 to 2000 ms, setting unit: 1 ms)
- When executing the low-speed execution type program, set either the constant scan time or low-speed program execution time.

(b) Set the execution type to "low-speed" in the parameter mode's program setting.

(c) Used for programs not necessary to be executed for every scan such as printer output.

(2) Multiple Low-Speed Execution Type Program

When multiple low-speed execution type program exists, the programs are executed in order of the smaller parameter mode setting number.

(3) Execution Time of the Low-Speed Execution Type Program to be Executed per Scan

(a) If there is an allowance time after completion of the operation of all low-speed execution type programs within one scan, the processing that will be executed after-that changes depending on the setting status of the special relay SM330 and the execution condition of the low-speed execution type program.

- Asynchronous system (SM330 = OFF): The operation of the low-speed execution type program is continued within the allowance time.
- Synchronous system (SM330 = ON) : If there is an allowance time, the operation of the low-speed execution type program is not continued and operation is started from the next scan.

Operation System of Low-Speed Execution Type Program	SM330 Setting Status	Execution Condition of Low-Speed Execution Type Program	
		When constant scan is set	When low-speed program execution time is set
Asynchronous system	OFF	Low-speed execution type program is re-executed*1	Low-speed execution type program is re-executed*2
Synchronous system	ON	Constant scan waiting time occurs*3	Operation of scan execution type program is started*4

*1 When the constant scan time is set, the low-speed execution type program is executed for the constant scan allowance time.
Therefore, the low-speed execution type program execution time differs for every scan. When the constant scan allowance time is less than 20 ms, the low-speed execution type program cannot be executed.
When using the low-speed execution type program, set the constant scan time so that the allowance time will be more than 2 ms.

*2 When the low-speed program execution time is set, the low-speed execution type program is executed for the set low-speed program execution time.
For this reason, the scan time differs for every scan.

*3 If the constant scan time has been set, the allowance time after completion of low-speed END processing is the waiting time, and when the preset constant scan time is reached, the scan execution type program is executed.

Constant scan waiting time
= (constant scan set time) - (scan time)
- (low-speed scan time)

Hence, the scan time per scan is constant.
However, if the constant scan allowance time is 2 ms or less, the low-speed execution type program cannot be executed. When using the low-speed execution type program, set the constant scan time so that the allowance time is more than 2 ms.

*4 When the low-speed program execution time has been set, the allowance time after completion of low-speed END processing is ignored and the operation of the scan execution type program is started.

Allowance time of low-speed program execution time
<Ignored>
= (set time of low-speed program execution time) - (low-speed scan time)

Hence, the scan time changes per scan.

(b) When the low-speed execution type program cannot be processed within the constant scan allowance time or low-speed execution program execution time, the program execution interrupts once, and the program is reset in the next scan.

POINTS

(1) Refer to "QnACPU PROGRAMMING MANUAL (Fundamentals)" for the index register processing when switching from the scan execution type program to low-speed execution type program.

(2) Refer to "QnACPU Programming Manual (Fundamentals)" for the index register processing when interrupt program is executed while executing the low-speed type program.

(3) Set the value of the low-speed program execution time so that the total of the low-speed program execution time and the scan time will be less than the WDT set value.

(4) The COM instruction cannot be used in a low-speed program.

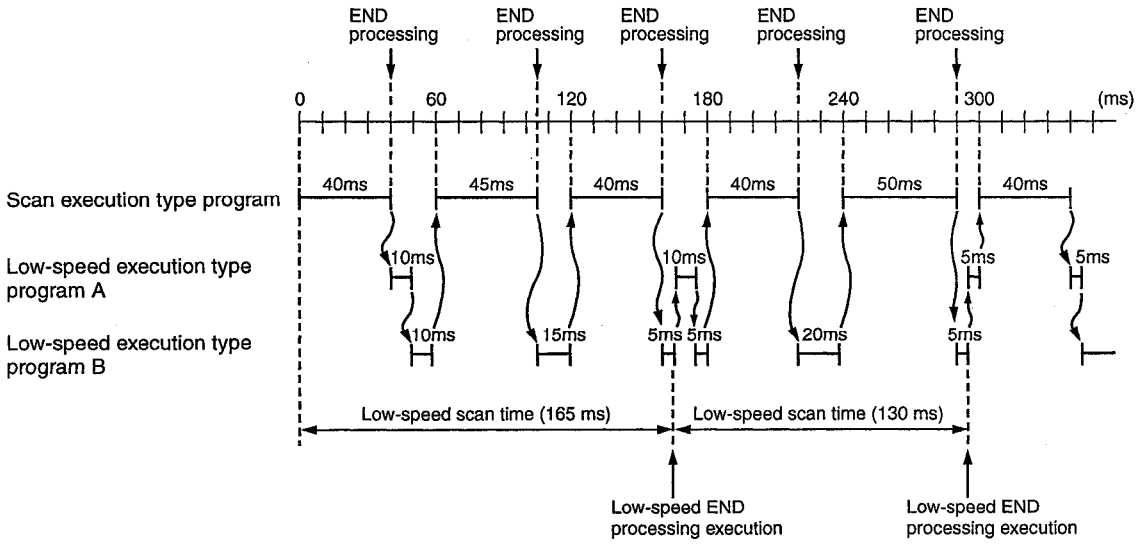
(5) When the "constant scan time" and "low-speed program execution time" have been set, PRG TIME OVER (error code: 5010) will occur if the (constant scan allowance time) becomes less than the (low-speed program execution time).
Execute the low-speed execution type program using either the constant scan time or low-speed program execution time.

1: Asynchronous system

(1) When setting the constant scan

The operation when the low-speed execution type program is executed with the following conditions is shown below:

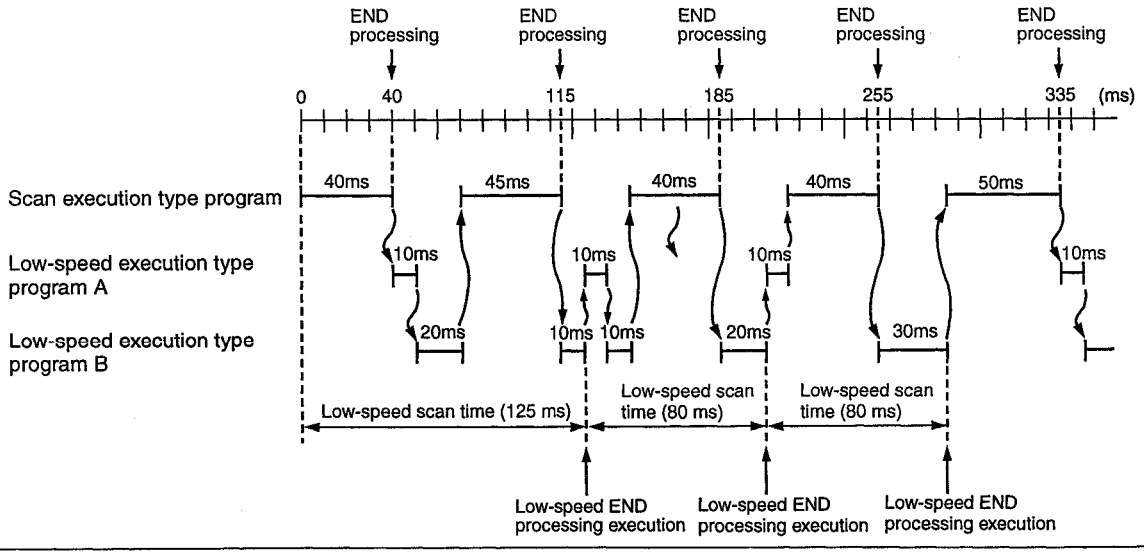
- Constant scan time : 60 ms
- Scan execution time program total : 40 ms to 50 ms
- Low-speed execution type program A execution time : 10 ms
- Low-speed execution type program B execution time : 30 ms
- END processing : 0 ms (assumed as 0 ms to simplify this example)
- Low-speed END processing : 0 ms (assumed as 0 ms to simplify this example)



(2) When setting the low-speed program execution time

The operation when the low-speed execution type program is executed with the following conditions is shown below:

- Low-speed program execution time : 30 ms
- Scan execution type program total : 40 ms to 50 ms
- Low-speed execution type program A execution time : 10 ms
- Low-speed execution type program B execution time : 30 ms
- END processing : 0 ms (assumed as 0 ms to simplify this example)
- Low-speed END processing : 0 ms (assumed as 0 ms to simplify this example)

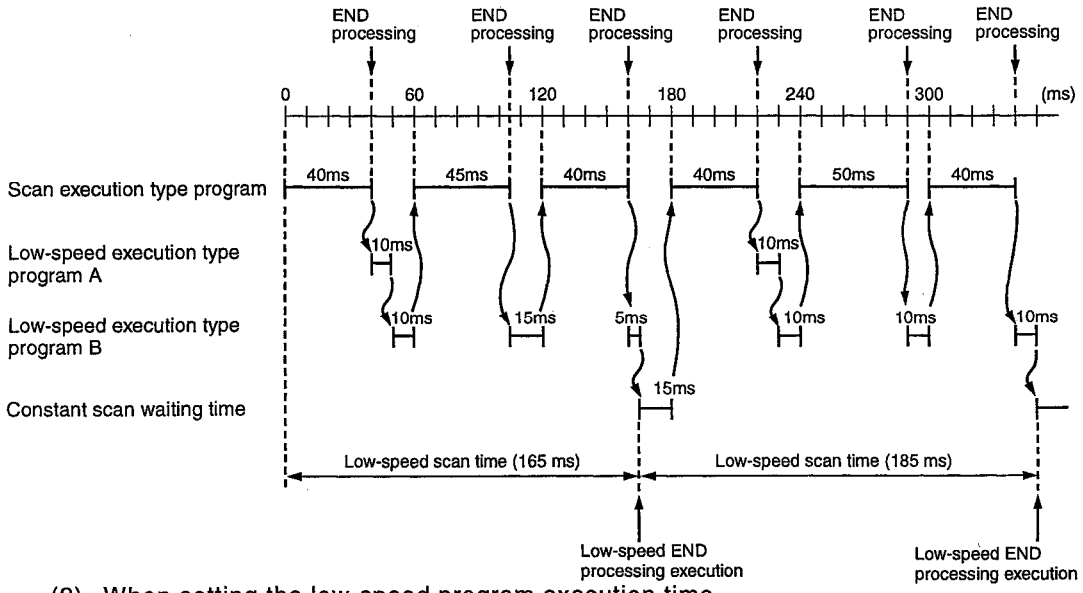


2: Synchronous system

(1) When setting the constant scan

The operation when the low-speed execution type program is executed with the following conditions is shown below:

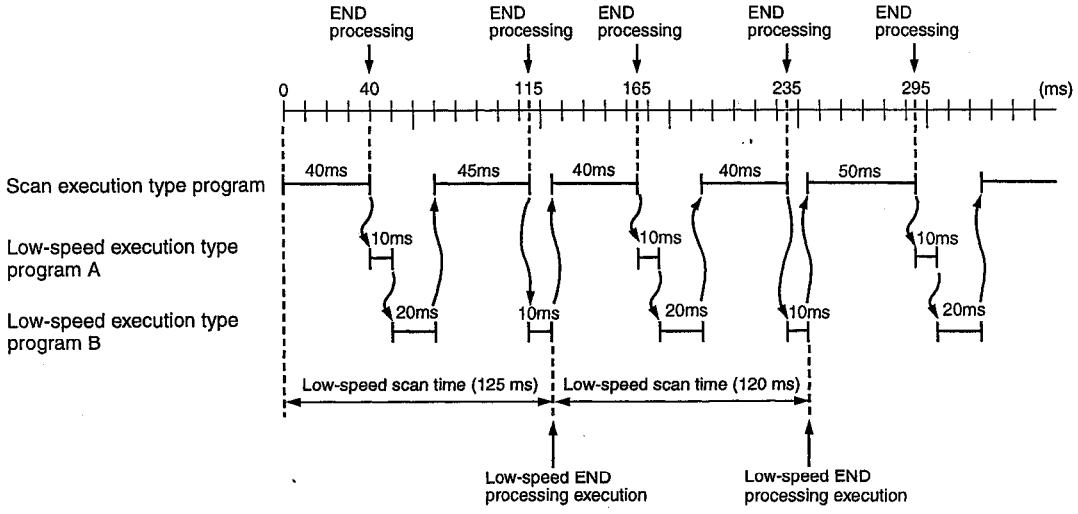
- Constant scan time : 60 ms
- Scan execution type program total : 40 ms to 50 ms
- Low-speed execution type program A execution time : 10 ms
- Low-speed execution type program B execution time : 30 ms
- END processing : 0 ms (assumed as 0 ms to simplify this example)
- Low-speed END processing : 0 ms (assumed as 0 ms to simplify this example)



(2) When setting the low-speed program execution time

The operation when the low-speed execution type program is executed with the following conditions is shown below:

- Low-speed program execution time : 30 ms
- Scan execution type program total : 40 ms to 50 ms
- Low-speed execution type program A execution time : 10 ms
- Low-speed execution type program B execution time : 30 ms
- END processing : 0 ms (assumed as 0 ms to simplify this example)
- Low-speed END processing : 0 ms (assumed as 0 ms to simplify this example)



(4) END Processing

The low-speed END processing is performed when all the low-speed execution type programs are executed.

The following processing is performed for the low-speed END processing.

- The low-speed program special relay/special register setting
- Low-speed execution program RUN write
- Low-speed scan time measurement
- Low-speed execution type program watchdog timer reset

When the low-speed END processing is complete, the low-speed execution type program is executed from the beginning again.

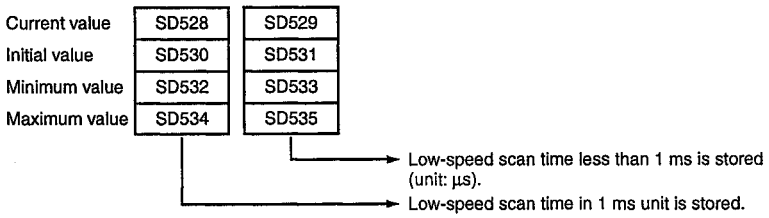
POINT

When the low-speed execution type program is being executed, "max. processing time for the instruction being executed + low-speed END processing time constants scan" may differ

(5) Low-Speed Scan Time

(a) The low-speed scan time is the total time of the time until the low-speed execution type program execution is complete and low-speed END processing time.
When a multiple low-speed execution type programs are executed, it is the total time of the time until the low-speed execution type program execution is complete and low-speed END processing time.

(b) The Q2ASCPU measures the low-speed scan time, and stores them in the special registers (SD528 to SD535). *1
By monitoring SD528 to SD535, the low-speed scan time can be checked.



[Example]

When 3 is stored in SD528 and 400 is stored in SD529, the low-speed scan time is 3.4 ms.

(6) Low-speed Execution Monitoring Time

This is a timer to monitor the low-speed execution type program execution time. The default value is not set.

When monitoring the execution time of the low-speed execution type program, it can be set in the range of 10 ms to 2000 ms in the parameter mode "PC RAS Setting". (Unit: 10 ms)

When the low-speed scan time exceeds the set low-speed execution monitoring time, it is set to "PRG TIME OVER" however the Q2ASCPU continues operations.

POINTS

(1) *1: The precision of each scan time stored in the special register is ± 0.1 ms.
The sequence scan time measurement continues even if the watchdog timer reset instruction (WDT) is executed in the sequence program.

(2) The low-speed execution monitoring time measurement is performed in the low-speed END processing.
So, if the low-speed execution monitoring time(t) is set to 100 ms and the low-speed scan time measured with the low-speed END processing exceeds 100 ms yields "PRG TIME OVER".

12.1.4 Stand-by Type Program

(1) What is a stand-by type program

(a) The stand-by type program is a program that is executed only when there is an execution request.

(b) The stand-by type program can be used in the following manner:

1) Program Library

The subroutine program and interrupt program can be set to stand-by type program and be separately maintained from the main program.

2) Program Process Change

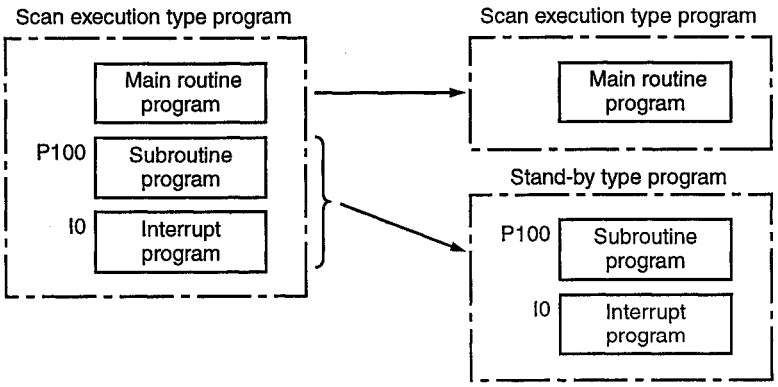
The main routine program is registered in the stand-by type program, the necessary program is changed to the scan execution type program, perform controls, and change the stand-by type program to the program not used in the program.

(2) For the Program Library

(a) Program Library

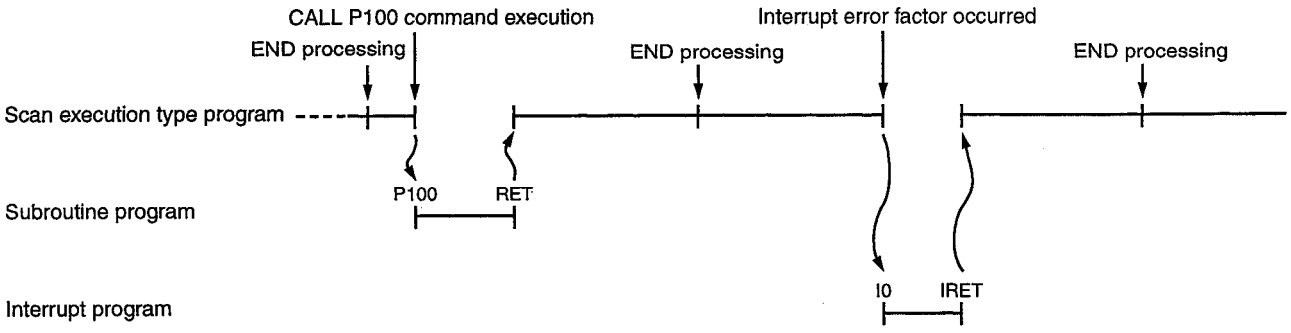
1) This is used when the subroutine programs and interrupt programs are separately maintained from the main routine.

Multiple interrupt programs with subroutine programs can be created in the same stand-by type program.



2) Upon completion of program execution, it returns to the program before the stand-by type program execution.

The operation when the stand-by type program subroutine and interrupt program are executed is shown below.



POINTS

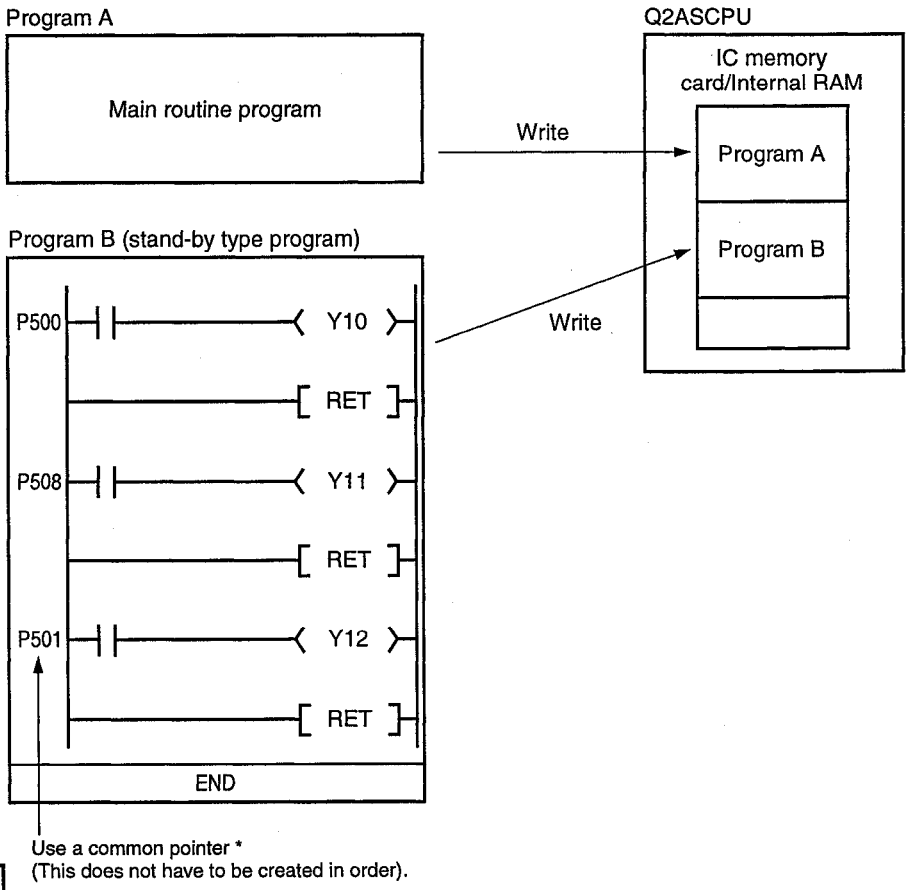
(1) The timer cannot be used for the stand-by type programs because current value updates and contact on/off are performed during the OUT T ⬚ instruction execution.

(2) Use the common pointer when a subroutine program is to be used as a stand-by type program.

Refer to "QnACPU PROGRAMMING MANUAL (Fundamentals)" for details on common pointers and local pointers.

(b) To Group a Subroutine Program as One Group

1) Create a subroutine program starting from step 0 of the stand-by type program. The END instruction is necessary at the end of the subroutine program.
2) There are no limitations to the order in which the subroutine program is created, so there is no need to create the program in the order of the pointer when a multiple subroutine program is created.
3) The pointer uses a common pointer.*
A common subroutine program that uses a common pointer can be called by all programs that are executed by Q2ASCPU.

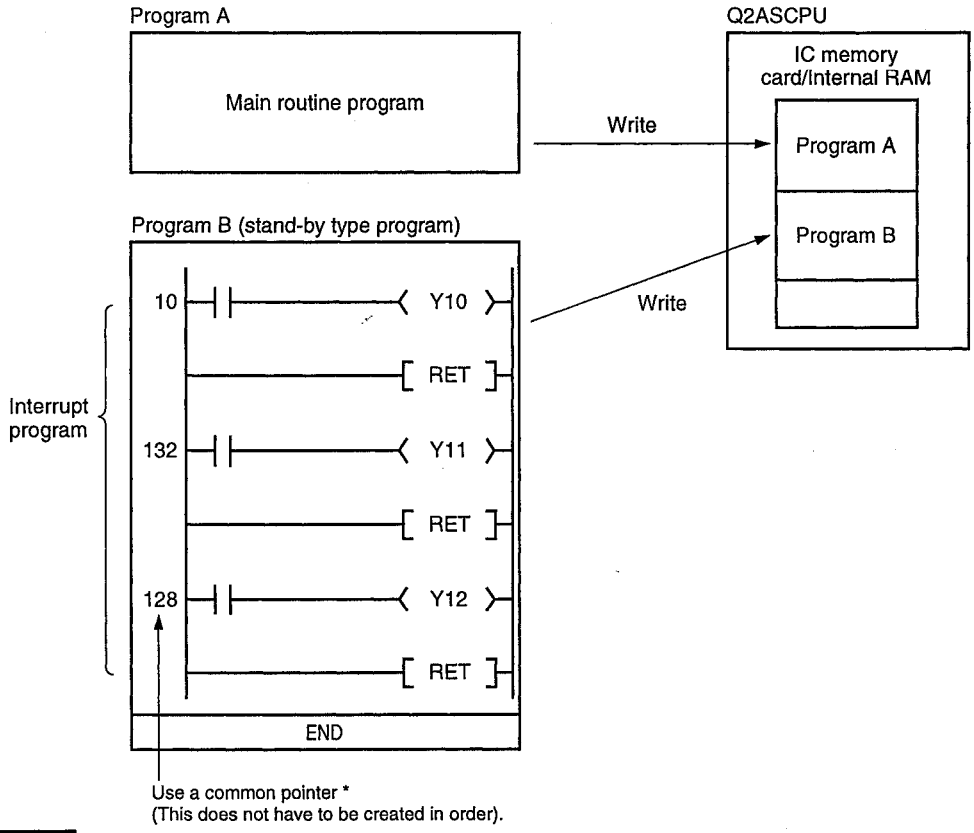


Remarks

*: Refer to "QnACPU PROGRAMMING MANUAL (Fundamentals)" for details on common pointers.

(c) To Group Interrupt Programs as One Program

1) Create the interrupt program from step 0 of the stand-by time program. The END instruction is necessary at the end of the interrupt program.
2) There are no limitations to the order in which the interrupt programs are created, so there is no need to create the programs in the order of the pointer when multiple interrupt programs are created.



Remarks

Refer to "QnACPU PROGRAMMING MANUAL (Fundamentals)" for details on interrupt pointers.

(3) To Switch the Program Procedure

(a) Create a program that is compatible with all programs and use it only to execute necessary programs.

Programs set by the parameter for stand-by type can be changed to a scan type program by a sequence program, and may be executed.

Changing the execution program by Q2ASCPU can be performed by the following instructions:

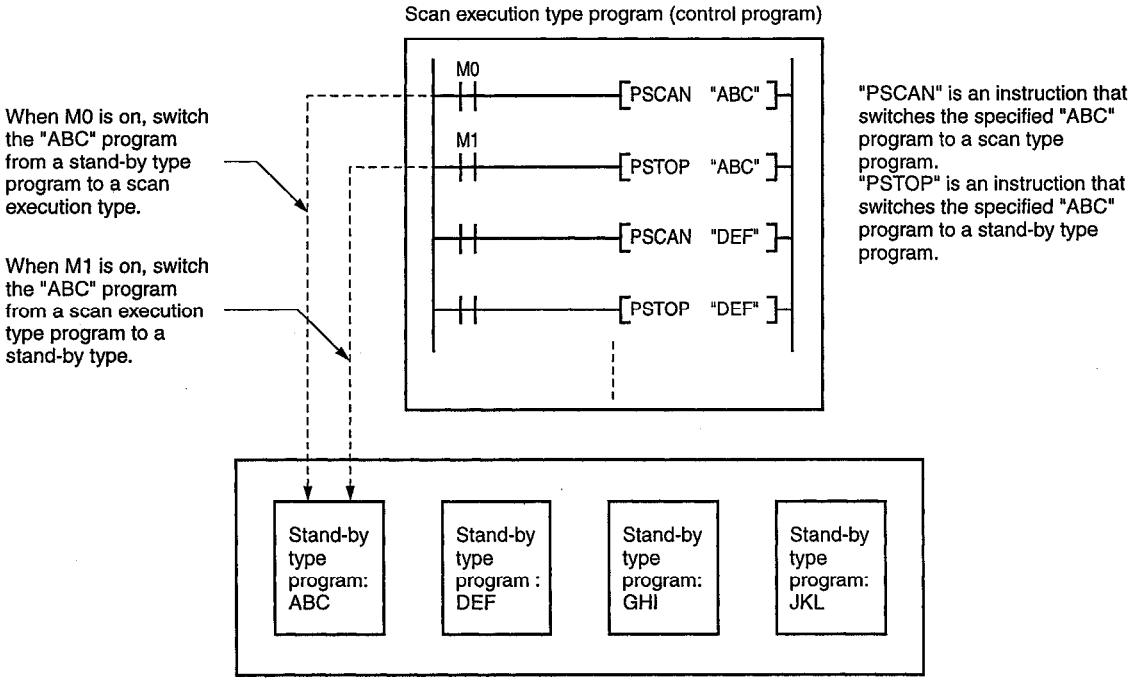
1) PSCAN instruction : Used to switch from a stand-by type program to a scan-execution program.
2) PLOW instruction : Used to switch from a stand-by type program to a low-speed program.
3) PSTOP instruction : Used to switch from a scan-execution program/low-speed type program to a stand-by type program.

(b) The following methods are available to switch execution programs.

1) To select a program that is executed by one control program.

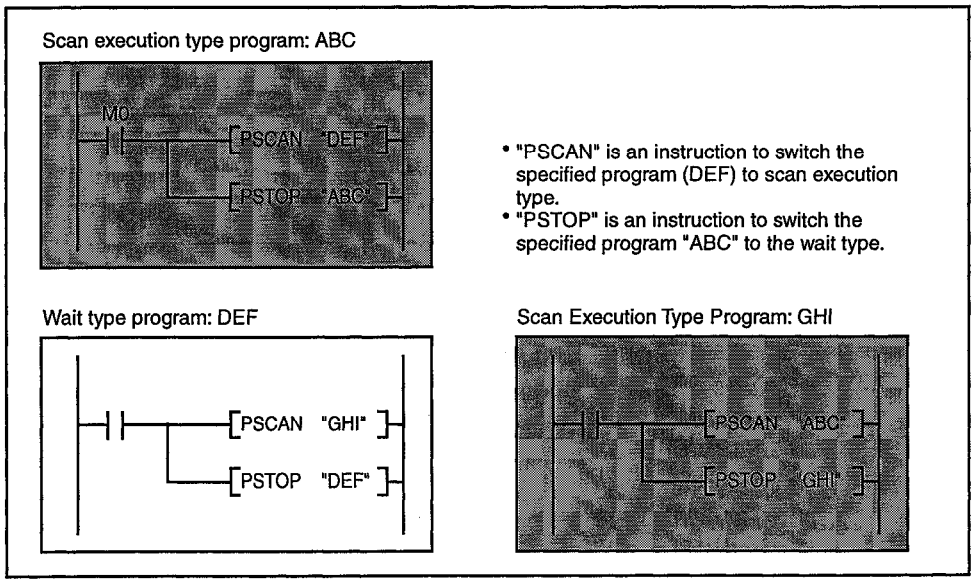
- Use the scan execution type program as a control program, and execute by changing the stand-by type program that meets the set criteria to a scan execution type program.

  Additionally, an unused scan execution type program can be changed to a stand-by type program.
- "ABC, "DEF," "GHI," and "JKL" stand-by program execution types can be changed by the management program shown by the following operations:

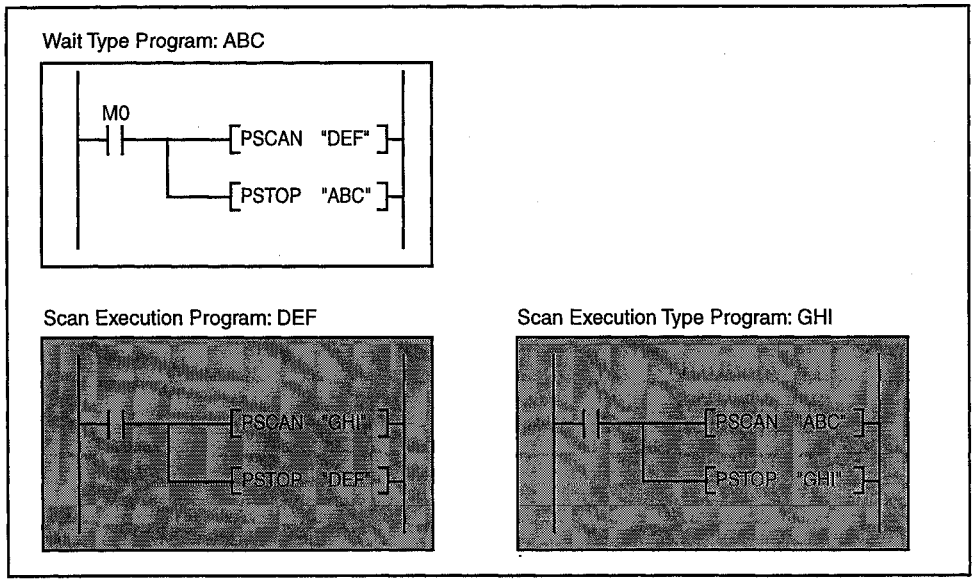

Scan execution type program (control program)
M0
PSCAN "ABC"
M1
PSTOP "ABC"
PSCAN "DEF"
PSTOP "DEF"
When M0 is on, switch the "ABC" program from a stand-by type program to a scan execution type.
When M1 is on, switch the "ABC" program from a scan execution type program to a stand-by type.
"PSCAN" is an instruction that switches the specified "ABC" program to a scan type program.
"PSTOP" is an instruction that switches the specified "ABC" program to a stand-by type program.
Stand-by type program: ABC
Stand-by type program : DEF
Stand-by type program: GHI
Stand-by type program: JKL

2) When changing to the execution type of another program from the scan execution type program being executed
   - Change the program to be executed next from the stand-by type to scan execution type with the scan execution type program being executed.
   - Set "ABC" and "GHI" programs to the scan execution type, and "DEF" program set to the stand-by type. The following shows the operation to switch the "ABC" program and "DEF" program execution types.

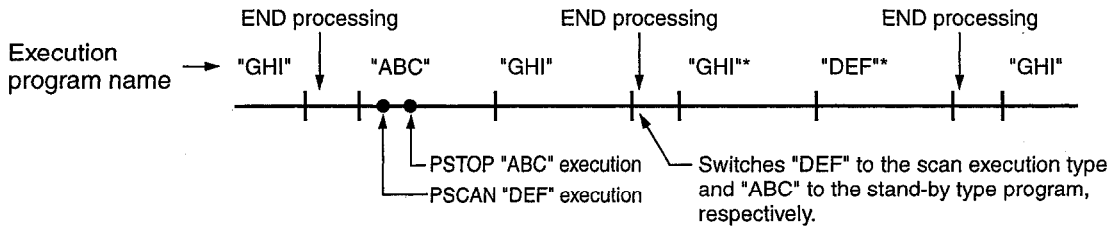
Before PSCAN, PTOP Instruction Execution


Scan execution type program: ABC
M0
PSCAN "DEF"
PSTOP "ABC"
• "PSCAN" is an instruction to switch the specified program (DEF) to scan execution type.
• "PSTOP" is an instruction to switch the specified program "ABC" to the wait type.
Wait type program: DEF
PSCAN "GHI"
PSTOP "DEF"
Scan Execution Type Program: GHI
PSCAN "ABC"
PSTOP "GHI"

After PSCN, PSTOP Instruction Execution



(c) The switching of program execution type with the PSCAN and PSTOP instructions are performed at END processing.
The program execution type does not change while this program is being executed.



Remarks

The execution order of the programs "GHI" and DEF" are set with the parameter program setting.

12.1.5 Initial Processing

This is a preprocessing for sequence operation execution, and is performed only once when the PLC is powered on or the CPU module is reset or switched from STOP to RUN.

When the power is turned on/reset

(1) The I/O module is reset and initialized.
(2) The range not latched in the data memory is initialized (bit device is off and word device is set to 0).
(3) In one of the items to self-diagnose in the PC CPU, the items which are to be checked when the PLC power is turned on and the CPU module is reset. (Refer to Section 9.3.)
(4) The I/O address of the I/O module is automatically allocated, according to the standard base module or extension base modules loaded position.
(5) For the MELSECNET/10 control station or MELSECNET (II)/B master station, the network/link parameter data is set in the network/data link module, and network communication/data link communication is started.
(6) When boot specification is set, the specified file is sent from the memory card to the internal RAM.

When STOP → RUN Note 1

(1) Same as (4) above.
(2) Same as (5) above. If any parameter value is changed during STOP, operation is performed with the new parameter value.
(3) Same as (6) above.
(4) When the device initialization value file is set, the device initialization values set in the file is set to the device.

Note 1. If any parameter value change or program modification is made during STOP, the RUN LED flickers when the key switch is moved from STOP to RUN, but the CPU module remains in a STOP status. (Refer to Section 15.3.)
Moving the key switch from STOP to RUN again places the CPU module in a RUN status, at which point the STOP → RUN-time operation is performed.

12.1.6 I/O Module Refresh Processing

The I/O module refresh processing is performed.
(Refer to the "QnACPU PROGRAMMING MANUAL (Fundamentals)".)

12.1.7 END Processing

This is a post-processing to return the sequence program execution to step 0 after completing the whole sequence program operation processing once.

(1) Performs fuse shutoff, I/O module verification, and battery low self-diagnosis. (Refer to Section 9.3.)

(2) When there is a data read/write request from the peripheral device, intelligent special function module, computer link module, serial communication module, Ethernet module or etc., the data exchange is performed with the PC CPU, peripheral device, and intelligent special function module.

(3) When a refresh request is made from the network/link module, refresh processing is performed.

(4) When the trace point of the sampling trace is at every scan (after END instruction execution), the set device status is stored in the sampling trace area.

(5) The refresh processing with the MELSECNET/MINI-S3 automatic refresh function is performed. (Refer to Chapter 7.)

POINTS

- When the constant scan function (refer to Section 10.2) is set, END processing time result is stored for the interval after END processing to the next scan.
- When executing the low-speed program (refer to Section 12.1.3), the low-speed END processing besides the normal END processing is performed.

  The low-speed END processing performs the setting of the low-speed program special relays and special registers.

12.2 RUN, STOP, PAUSE, STEP-RUN Operation Processing

The Q2ASCPU has four types of operation states; RUN, STOP, PAUSE, and STEP RUN states. The PC CPU operation processing is explained below:

(1) RUN Status Operation Processing
  (a) RUN status is when the sequence program operation is performed from step 0 → END (FEND) instruction → step 0 repeatedly .
  (b) When entering the RUN state, the output state saved at STOP by the parameter output-mode setting during STOP → RUN.
  (c) The processing time of switching from STOP → RUN until the beginning of sequence program operation changes with system configurations, but usually is 1 to 3 seconds. However, this time may be longer depending on the conditions.

(2) STOP Status Operation Processing
  (a) STOP status is when the sequence program operations are stopped with the RUN/STOP key switch or remote STOP (refer to Section 10.6.1) is performed.
  (b) When entering the STOP state, save the output state and turn off all output. The data memory other than output (Y) are maintained.

(3) PAUSE Status Operation Processing
  (a) The PAUSE state is when the sequence program operations are paused while maintaining the output and data memory status. (Refer to Section 10.6.3.)

(4) STEP-RUN Operation Processing
  (a) STEP-RUN is a mode where the sequence program operation processing can be stopped and continued per instruction using the GPP function. (Refer to Section 8.7.)
  (b) The operation processing is interrupted while maintaining the output and data memory, so the execution status can be checked.

(5) Q2ASCPU Operation Processing with RUN/STOP Key Switch Operation

Q2ASCPU operation processing / RUN/STOP key switch operation	Sequence program operation processing	External output	Data Memory (Y, M, L, S, T, C, D)	Remarks
RUN → STOP	Executes up to the END instruction and stops.	OS saves the output state and all output are off.	Maintains the status immediately before the STOP state.	
STOP→ RUN	Starts.	Determined by the output mode of the parameter at STOP → RUN.	Starts executing the operation from the status immediately before the STOP state.	

POINTS

The Q2ASCPU performs the following in any of RUN, STOP, and PAUSE states;

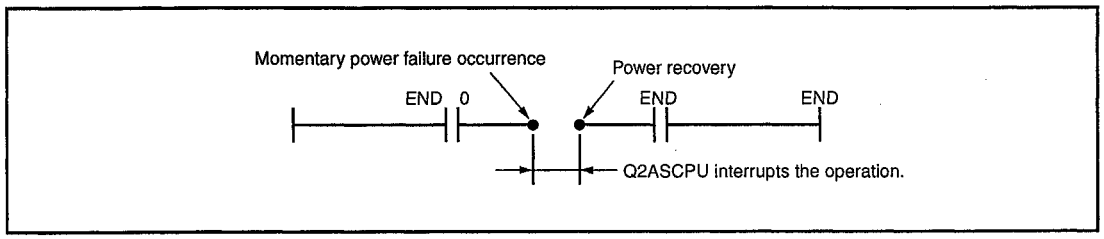
- I/O module refresh processing
- Data communication with the peripheral device, computer link module, and serial communication module
- Link refresh processing

Therefore monitoring and/or test operation of I/O from peripheral device, read/write from the computer link module or serial communication and communications with other stations can be performed with MELSECNET.

12.3 Operation Processing during Momentary Power Failure

The Q2ASCPU detects a momentary power failure to the power module when the input power voltage is lower than the regulated ranges.
When the Q2ASCPU detects a momentary power failure, the following operation processing is performed:

(1) When momentary power failure occurs for less than permitted power failure time
   (a) The output is maintained when the momentary power failure occurs, and file name of the file accessed and error history are logged. Then the system interrupts the operation processing. (The timer clock continues.)
   (b) When there is an SFC continue specification, a system saving processing is performed.
   (c) When a momentary power failure occurs, the operation processing is continued.
   (d) Even if the operation is interrupted due to momentary power failure, the watchdog timer (WDT) measurement continues. For example, if the GPP function parameter mode WDT setting is set at 200 ms, when a momentary failure of 15 ms occurs at scan time 190 ms, the watchdog error is set.



(2) When a power failure occurs for more than the permitted power failure time
   Q2ASCPU is reset.
   When the reset operation is performed by turning the PLC power on or by performing reset operation using the RUN/STOP key-switch of the CPU module, the same operation processing is performed.

12.4 Data Clear Processing

The Q2ASCPU clears all data except for the following, when a reset operation is performed with a RUN/STOP key switch RUN/STOP → RESET, or power ON → OFF → ON.

(1) Internal RAM data (except for when there is a memory clear specification at boot specification.)*1
(2) Data in the memory card.
(3) Device data with latch specification
(4) File register data
(5) Local device data
(6) Failure history data (when special relay SD storage)

*1 Refer to "QnACPU PROGRAMMING MANUAL (Fundamentals)" for boot specifications.

Data in (3) and (6) are cleared using the RUN/STOP key switch latch clear operation (Refer to Section 15.3) or remote latch clear operation (refer to Section 10.6.5) from the GPP function. The method to specify device latch is to set the latch range for each device in the GPPQ parameter mode on the "device setting" screen. There are two types of latch range settings:

(1) Valid latch clear key : Sets the latch range that can be cleared with the STOP/RUN key-switch latch-clear operation.
(2) Invalid latch-clear key : Sets the latch range that cannot be cleared even with the STOP/RUN key-switch latch-clear operation.

The devices that were set to latch-clear key can only be cleared by an instruction or GPP function.

Instruction to clear method	:	Reset with the RST instruction or send K0 with the MOV instruction.
GPP function clear method	:	Clears all device memory in the online mode for PC menu.

Refer to "QnACPU PROGRAMMING MANUAL (Fundamentals)" for details of the device latch range.
Refer to the "GX Developer OPERATING MANUAL" or "SW□IVD-GPPQ OPERATING MANUAL(Online/Offline)" for details of the GPP function operation methods.

MEMO